NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

PREVISÕES DE MOVIMENTOS ACENTUADOS NO PREÇO DE AÇÕES COM SELEÇÃO AUTOMATIZADA DE MODELOS DE APRENDIZAGEM AUTOMÁTICA

por

Gil Gonçalo Freire Martins

Trabalho de Projeto apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão do Conhecimento e Business Intelligence

Orientador: Roberto Henriques

Fevereiro 2018

RESUMO

A previsão de séries temporais relativas a mercados financeiros é tida como uma tarefa desafiante. Prever o movimento e direção das séries pode ser mais lucrativo que o prever preço, mas de forma a maximizar o retorno pode interessar prever movimentos acentuados das séries. Fazendo uso de 18 séries de ações provenientes de três mercados distintos e para um horizonte de previsão de 4 semanas, este estudo testa três hipóteses para avaliar a pertinência da previsão de movimentos acentuados, considerando a existência, ou não, de dependência na dimensão tempo e também a potencial alteração das relações entre as variáveis dependentes e independentes com a dimensão tempo. Para cada série são testadas, as três hipóteses, com sete modelos de aprendizagem automática e selecionado o melhor para realizar as previsões através de uma metodologia de otimização automática, que seleciona de entre 103 variáveis possíveis para realizar as previsões, num problema com classes desequilibradas. Este estudo atinge resultados competitivos com a literatura, obtendo-se com a melhor metodologia uma probabilidade de acertar na direção do movimento da série de 80% e num movimento acentuado de 65% das vezes.

PALAVRAS-CHAVE

Previsão movimento ações; classes desequilibradas; seleção variáveis; aprendizagem automática; deriva conceito

ABSTRACT

Forecasting time series on financial markets is regarded as a challenging task. Predicting the movement and direction of the series may be more profitable than the price forecast, but in order to maximize the return it may be interesting to predict the series sharp movements. Using 18 series of stocks from three different markets and for a forecast horizon of 4 weeks, this study tests three hypotheses to evaluate the pertinence of prediction of sharp movements, considering the existence or not of dependence in the time dimension and also the potential change of the relationships between the dependent and independent variables with the time dimension. For each tested series, the three hypothesis, with seven machine learning models are tested and the best is selected to carry out the predictions through an automatic optimization methodology, which selects from 103 possible variables to do the predictions, in a problem with unbalanced classes. This study achieves competitive results with the literature, with the best methodology being likely to hit in the direction of the movement of the series in 80% of the cases and the sharp movements in 65% of times.

KEYWORDS

Forecast stock movement; unbalanced classes; feature selection; machine learning; concept drift

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ADASYN	*Adaptative Synthetic Sampling*
ANN	*Artificial Neural Network*
AUC-PR	Area sob a curva de precisão-revocação
DOB-SCV	*Distribution optimally balanced stratified cross-validation*
FCBF	*Fast Correlation Based Filter*
GBM	*LightGBM*
HER	Hipotese das Expectativas Racionais
HME	Hipotese do Mercado Eficiente
KNN	*K-Nearest Neighbours*
LOG	*Logistic Regression*
MLP	*Multi-layer Perceptron*
mRMR	*Minimum redundancy maximum relevance*
NB	*Gaussian Naive Bayes*
PR	Curva precisão-revocação
RBF	*Radial Basis Function Kernel*
RF	*Random Forest*
ROC	*Receiver operating characteristic curve*
SENN	*Synthetic Minority Oversampling Technique followed by Edited Nearest Neighbours*
SMOTE	*Synthetic Minority Oversampling Technique*
SNR	Relação Sinal Ruido
SVM	*Support Vector Machine*
VC	Validação Cruzada

1. INTRODUÇÃO

A previsão de mercados financeiros é uma tarefa desafiante, porque este género de séries apresentam geralmente um comportamento não-linear, complexo e volátil, influenciado por múltiplos fatores, como eventos políticos, expectativas dos intervenientes ou até mesmo catástrofes naturais.

Vários estudos foram efetuados relativamente à hipótese do mercado eficiente (HME), que teoriza que os mercados financeiros não são possíveis de serem previstos. De acordo com esta hipótese os mercados seguem um passeio aleatório, determinados racionalmente com base em toda a informação disponível no momento e como tal eficientes (Fama, 1965).

Relacionada com a HME encontra-se ainda a Hipótese das Expectativas Racionais (HER), que postula a incapacidade de qualquer algoritmo preditivo ser melhor que os participantes no mercado, utilizando toda a informação disponível. Ou seja, que os participantes possuem as mesmas expectativas homogéneas sobre o futuro. No entanto, Shostak (1997) refere que a negociação no mercado implica expectativas heterogéneas, ou seja, enquanto compradores esperam uma subida do mercado, os vendedores esperam uma descida. Mesmo quando todos têm igual acesso a informação similar como notícias e outra informação pública, existe sempre o problema da interpretação distinta da mesma.

Ainda, segundo Degutis e Novickyté (2015), a HME, falha em explicar ineficiências nos mercados como o excesso de volatilidade, a reação desproporcional dos investidores ou a sazonalidade dos mercados. Grossman (1975) apresenta mesmo um paradoxo, que refere que se é consenso geral de que o mercado é eficiente, então os participantes deixariam de recolher e analisar a informação, pela suposta inutilidade da mesma, o que geraria ineficiência.

Leung, Daouk e Chen (1999) concluem que negociar no mercado através da previsão da direção dos movimentos das ações pode ser mais lucrativo que prever o valor exato do preço das séries financeiras. Desta forma, utilizando diversos algoritmos supervisionados, e indicadores técnicos derivados do preço de fecho do mercado e do volume de transações ocorridas como entrada dos mesmos, vários trabalhos recentes reportam sucesso a prever a direção do movimento de séries financeiras utilizando dados diários, codificando para o efeito uma variável dependente binária, onde 1 indica uma subida e 0 uma descida do mercado para determinado horizonte temporal futuro. Contudo parece não existir um algoritmo ou indicadores técnicos que dominem a performance nos diversos estudos.

O teorema "*No Free Launch*" demonstrou que os algoritmos de aprendizagem automática não são obrigatoriamente universalmente adequados (Wolpert & Macready, 1997). Assim um algoritmo ou variáveis específicas não se comportam da mesma forma em diferentes séries financeiras. Torna-se então pertinente automatizar o processo de seleção de modelos e variáveis para prever os movimentos de diversas ações com reduzida intervenção humana.

Por outro lado, Harries e Horn (1995) referem que alterações de taxas de juro, eventos políticos e mundiais ou taxas de inflação podem afetar as distribuições dos dados das séries

financeiras. A este comportamento dá-se o nome de deriva de conceito[1], ou seja, quando a relação entre as variáveis independentes e dependentes altera com a dimensão tempo (Dumasia & Shah, 2016; Hoens, Polikar, & Chawla, 2012; Webb, Hyde, Cao, Nguyen, & Petitjean, 2016) . Mittal e Kashyap (2016) alegam mesmo que no mundo real, a maioria dos problemas são dinâmicos por natureza e não estacionários. Este é um desafio, porque a maioria dos modelos tradicionais como *Suport Vector Machines* (SVM) ou *Artificial Neural Networks* (ANN) assumem que os dados, para teste e validação, são obtidos da mesma distribuição (X. Yu, Yu, Xu, Yang, & Xie, 2015). Em muitos dos casos, os dados são mesmo obtidos de forma incremental ou em blocos com o decorrer do tempo, tornando-se desafiante utilizar todos os dados até determinado momento para prever os subsequentes. Desta forma utilizar uma metodologia apropriada e eficiente para lidar com a deriva de conceito pode oferecer vantagens competitivas na previsão de séries financeiras.

Ademais, numa situação onde o capital disponível para investimento é limitado interessa tentar maximizar o retorno do investimento, negociando apenas quando se prevê movimentos acentuados nas séries. E não simplesmente quando se prevê uma subida ou descida do preço. No entanto, esta codificação das classes, num limite superior, levará à criação de um problema não balanceado, e ao que se conseguiu averiguar, não abordado na literatura, na previsão do movimento de séries financeiras. Por outro lado, considerar um limite superior poderá também reduzir o risco de negociar no mercado, quando o modelo prevê um falso positivo, este poderá mesmo assim, com determinada probabilidade, traduzir-se em ganhos.

Para testar as hipóteses anteriores, este estudo cria uma metodologia automatizada para selecionar de entre 103 indicadores técnicos distintos a apresentar a diferentes algoritmos de aprendizagem automática cujos híper-parâmetros são otimizando automaticamente através de validação cruzada.

Testamos três metodologias diferentes para um horizonte de previsão de 4 semanas. Consideramos no treino dos modelos que existe dependência temporal entre os dados, assim como o contrário, alterando respetivamente a forma como se efetua a partição dos dados no processo de validação cruzada utilizado, devido ao facto de existir bibliografia que considera ambas as hipóteses, reportando resultados satisfatórios. Para testar a pertinência da utilização de algoritmos que tenham em conta a deriva de conceito aplicamos e avaliamos o algoritmo Learn^{++}.NIE (Ditzler & Polikar, 2012), que se apresenta como o algoritmo estado da arte para problemas não balanceados e com deriva de conceito.

Assim, este estudo contribui para o aprofundar do conhecimento, avaliando diferentes metodologias para prever o movimento de ações, num contexto não balanceado, e considerando a existência de deriva de conceito, que ao que se conseguiu apurar não foi ainda abordado na literatura. Implementando uma metodologia automatizada para cada série financeira, de forma a selecionar o modelo com melhor performance para prever o movimento da mesma, otimizando dinamicamente os híper-parâmetros dos modelos e a seleção das variáveis de entrada. Avaliando ainda a aplicabilidade dos resultados das previsões para negociar no mercado, utilizando probabilidades de forma a gerir o risco.

Este documento encontra-se organizado da seguinte forma: na secção 2 é discutida a bibliografia relevante aos desafios supracitados; na secção 3 apresenta-se a metodologia e modelos

[1] Do inglês *Concept Drift*

utilizados neste estudo, analisados empiricamente na secção 4. Finalmente na secção 5 apresenta-se as principais conclusões e limitações do trabalho desenvolvido assim como propostas de trabalho futuro.

2. REVISÃO DA LITERATURA

Nesta secção revemos a literatura referente aos desafios introduzidos na secção anterior deste trabalho. Na secção 2.1 discutiremos a abordagem que outros trabalhos seguiram para prever a a direção e movimento das séries financeiras. Na secção 2.2, devido à codificação que realizaremos à variável dependente, revemos na literatura, as técnicas e desafios inerentes à previsão com dados não balanceados. Abordaremos ainda o desafio da deriva de conceito associado a dados não balanceados. Por fim na secção 2.3 revê-se o disposto na literatura relativamente à seleção de variáveis para apresentar aos modelos de aprendizagem automática.

2.1. REVISÃO LITERATURA RELATIVAMENTE À PREVISÃO DA DIREÇÃO DO MOVIMENTO

Em diversos estudos realizados para diferentes mercados parecem verificar-se a rejeição da HME, sendo que a maioria apresenta resultados melhores que o investimento aleatório.

Vaiz e Ramaswami (2016) utilizam árvores de decisão e dados do mercado Indiano, reportando valores de exatidão na ordem dos 85%. Os autores referem ainda que efetuam a divisão dos dados utilizando 70% para treino e 30% para teste dos modelos. Patel et al. (2015) comparam quatro modelos, uma ANN, uma SVM, *Random Forest* (RF) e o modelo de *Naive-Bayes*, também no mercado Indiano, reportando valores de exatidão mínimos de 65% e máximos de 91% em diversas séries utilizando os diferentes modelos. De entre os quais os autores referem que o modelo Naive-Bayes é o que apresenta menor exatidão nas séries testadas contrastando com as RF se traduzem nos melhores resultados. Estes últimos autores codificam variáveis de entrada de forma binária com base no conhecimento tácito existente sobre a informação contida nos indicadores, caso o indicador reporte uma tendência de descida ou subida.

Por sua vez, Khaidem, Saha e Dey (2016) utilizam uma RF e alisam exponencialmente os dados de fecho do mercado, antes de calcular os indicadores técnicos a apresentar ao modelo, numa tentativa de reduzir o ruído presente nas séries, para prever a direção de ações do mercado financeiro dos Estados Unidos da América (EUA), reportando uma exatidão mínima de 84% e máxima de 94% em três séries distintas. Dey et al. (2016) estudam também o mercado dos EUA aplicando um algoritmo de *boosting*[2] de Árvores de decisão (*Xtreme Gradient Boosting*) e reportam resultados de exatidão entre 87% e 99%, nas duas séries testadas e em diversos horizontes temporais, depois de também as alisarem exponencialmente. Não referindo qual o grau de alisamento ótimo.

No mercado da Turquia, Kara, Boyacioglu e Baykan (2011) testam SVM e ANN, para prever a sua direção futura, reportando uma exatidão média superior a 70%. Para prever a direção do movimento do mercado Iraniano Alavi, Sinaei e Afsharirad (2015) testam três modelos diferentes, de entre os quais uma RF que se destaca dos restantes com uma exatidão de 91% quando comparada a uma exatidão de 84% e 81 %, de uma SVM e um algoritmo *K-nearest Neighbours* (KNN) respetivamente.

[2] Método que ajusta classificações subsequentes a favor dos erros de classificação em classificações anteriores

Contudo nos estudos anteriores, Kara et al. (2011) e Patel et al. (2015) não respeitam a temporalidade dos dados na seleção dos conjuntos de treino e de teste. Os restantes estudos não detalham na metodologia para seleção de dados para os conjuntos de treino e teste, pelo que não é possível inferir se preservam a temporalidade dos mesmos. Não respeitar a temporalidade dos dados implica assumir que os dados nos dois conjuntos são independentes e identicamente distribuídos (i.i.d.), ou seja que não existe uma estrutura dependente entre os dados. No entanto, esta independência pode ser comprometida quando os dados são auto correlacionados, sendo obtidos para teste e treino perto no domínio temporal. Pelo que alocar aleatoriamente observações para realizar validação e teste do modelo não remove necessariamente a informação contida nas observações utilizadas para treino, podendo levar a resultados e conclusões extremamente otimistas na seleção dos modelos. Roberts et al. (2017) demonstra o anterior, referindo que para efetuar previsões no futuro os conjuntos de dados de treino e teste devem respeitar a temporalidade e ser contíguos, simulando comportamento real.

Não obstante, Qiu e Song (2016) preveem o movimento para o próximo dia do índice Japonês, preservando a temporalidade dos dados, demonstrando a importância das variáveis selecionadas para treinar uma ANN otimizada por um Algoritmo Genético, apresentando performance para a exatidão do modelo de 61% e 81% dependendo do subconjunto de variáveis selecionado para treino. Por outro lado, também preservando a temporalidade dos dados, e utilizando uma SVM, Madge (2015) não obtém, com dados diários e em diferentes horizontes de previsão, resultados médios superiores a 60% para 34 ações tecnológicas no mercado dos EUA, melhorando estes, contudo, à medida que o horizonte de previsão considerado aumenta. Também preservando a temporalidade entre teste e treino e utilizando 12 indicadores técnicos como entradas, Kumar e Thenmozhi (2005), testam cinco modelos em dados diários, concluindo que uma SVM linear e uma RF apresentam os melhores resultados seguidos por uma ANN e um modelo logístico com uma exatidão de 68.44%, 67.40%, 62.93% e 59.60% respetivamente.

Vários dos estudos referidos apresentam também todos os indicadores técnicos escolhidos aos modelos para realizar previsões dos movimentos das séries sem efetuarem uma seleção analítica prévia dos mesmos, introduzindo assim, potencialmente, variáveis irrelevantes nos modelos, que podem levar a um sobre ajustamento aos dados e fraca generalização fora da amostra (Domingos, 2012; Li et al., 2016). Existem várias vantagens em realizar uma seleção prévia de variáveis a apresentar aos modelos, como facilitar a visualização e entendimento dos dados, reduzir os requisitos de armazenamento necessários, e importante no atual estudo, reduzir o tempo de treino dos modelos e combater a maldição da dimensionalidade para melhorar a performance das previsões (Guyon & Elisseeff, 2003). Shivaji et al. (2013) referem ainda que um número elevado de variáveis, torna difícil a um modelo aprender na presença de interações complexas e ruído, pelo que a adição de algumas variáveis pode reduzir a qualidade do modelo. Esta é mesmo considerada uma das etapas mais importantes num processo de prospeção de dados (Goswami, 2014).

Kara et al. (2011) referem que os indicadores técnicos escolhidos no estudo derivam da consulta bibliográfica que realizaram, utilizando os mais comuns em estudos sobre os mercados financeiros. Mas, pode ocorrer que, diferentes séries apresentam diferentes interações entre os indicadores técnicos e a variável dependente, pelo que assumir que as interações são as mesmas para todas as séries pode não ser correto e limitar os resultados obtidos.

Por outro lado, como já referido na secção 1, a possível existência de deriva de conceito pode prejudicar a aprendizagem dos algoritmos. Segundo Dumasia e Shah (2016) existem 4 tipos de deriva: 1) deriva abrupta, que altera todas as relações de forma abrupta; 2) deriva gradual que se caracteriza pela coexistência de relações ao longo de determinado período de tempo entre dois ou mais variáveis; 3) deriva recorrente que se verifica quando conceitos do passado reaparecem; 4) deriva incremental que ocorre com alterações uniformes ao longo do tempo.

Vários métodos foram propostos para lidar com deriva de conceito. Fazendo uso de janelas deslizantes, o algoritmo FLORA (Widmer & Kubat, 1996), descarta dados do passado, impedindo que padrões antigos sejam recuperados. O algoritmo DDM (Gama, Medas, Castillo, & Rodrigues, 2004) monitoriza constantemente alterações nas distribuições das variáveis, treinando um novo modelo de aprendizagem automática quando deteta mudanças, apresentando boa performance para lidar com derivas abruptas e derivas graduais, desde que estas ultimas ocorram rapidamente. O algoritmo EDDM (Bifet et al., 2006) é uma extensão do anterior, abordando o ponto fraco supracitado, e resolvendo-o, contudo não é também capaz de modelar deriva recorrente.

Os algoritmos anteriores são ativos na procura da deriva, necessitando de encontrar o ponto exato no tempo e intensidade da deriva, podendo gerar falsos positivos na procura de deriva afetando o algoritmo base de aprendizagem automática (Elwell & Polikar, 2011). Por sua vez, nas abordagens passivas os algoritmos aprendem continuamente, reforçando o conhecimento anterior caso não tenha ocorrido deriva e aprendendo a mudança quando ela ocorre.

Desta forma, outra abordagem, passiva, é o uso de agrupamentos de diferentes modelos, treinados em distintas janelas passadas. O algoritmo SEA (Street & Kim, 2001) cria um agrupamento de modelos substituindo o modelo treinado à mais tempo para adicionar novo, enquanto que o algoritmo DWM (Kolter & Maloof, 2007) substitui o que menos contribui, pelo que fazem uso de um numero fixo de modelos no agrupamento, perdendo assim informação do sistema, fazendo-os não ideais para lidar com deriva recorrente espaçada. Por sua vez algoritmo Learn^{++}.NSE proposto por Elwell e Polikar (2011) que é uma abordagem baseada em combinações de modelos treinados e com pesos dinamicamente atualizados ao longo do tempo, à medida que os conceitos alteram. É capaz de modelar dados não estacionários, como mudanças de conceito abruptas, graduais ou recorrentes, resultando num modelo robusto em dados recentes, que rapidamente, faz uso de dados do passado para explicar mudanças de conceitos detetadas. Contudo, devido a métrica utilizada, assume que os dados são balanceados (Ditzler & Polikar, 2012).

Na próxima secção, abordamos este assunto, quando associado a classes desequilibradas.

2.2. CLASSIFICAÇÃO COM DADOS NÃO BALANCEADOS

Geralmente, os algoritmos de aprendizagem automática assumem que o número de observações em cada classe é similar, existindo contudo situações onde tal não ocorre, causando dificuldades à aprendizagem dos dados e, fazendo com que estes apresentem resultados enviesados em direção à classe maioritária (Krawczyk, 2016). Ironicamente, a classe minoritária, é geralmente a de maior importância e interesse do ponto de vista da aprendizagem (Ali, Shamsuddin, & Ralescu, 2015; Elkan, 2001). No contexto dos mercados financeiros, obter falsos positivos pode mesmo possuir um custo associado maior que o de obter um falso negativo, uma vez que entrar no mercado

apoiado num falso positivo pode levar a perdas. Ou seja, o custo associado a um erro Tipo I quando comparado a um erro Tipo II pode revelar-se dramaticamente assimétrico.

Provost (2000) apresenta duas principais causas para este problema. O uso de métricas globais de performance, como a exatidão que podem dar vantagem à classe maioritária; e que o algoritmo assume que os dados utilizados originam da mesma distribuição.

Quando a tentativa de maximizar a exatidão da classificação é uma das métricas utilizadas, e as classes não são balanceadas, esta poderá fornecer informação errónea e enganadora sobre a performance de um modelo. Considere-se o caso onde existem 90 observações da classe maioritária e 10 da minoritária, um algoritmo com uma exatidão de 90%, pode unicamente ter aprendido a classe maioritária. De fato, alguns dos estudos na secção anterior reportam esta métrica, e apesar de codificarem as classes em subida e descida (limite de zero), isto não invalida a existência de dados não balanceados, dependendo da tipologia da série considerada.

Segundo, HE e Garcia (2010), métricas como a precisão, a sensibilidade, *F-score*, *G-mean* e a Curva característica de operação do recetor[3] (ROC) , são adotadas frequentemente em problemas que lidam com dados não balanceados. Contudo, segundo estes mesmos autores, embora mais apropriadas que a exatidão**,** estas são ineficientes ou, no caso da ROC, limitada para o efeito. Como alternativa os autores descrevem a área debaixo da curva de precisão-revocação (AUC-PR), como uma métrica apropriada para dados não balanceados, e recomendada por López, Fernández, García, Palade e Herrera (2013) quando a classe minoritária é importante. De acordo com Davis e Goadrich (2006), esta curva possui uma forte correspondência com a ROC**,** onde uma curva domina o espaço ROC se e só se dominar o espaço PR. Assim, se um algoritmo otimizar a área debaixo da curva ROC, não é garantido que otimize a área debaixo da curva PR. Desta forma, olhar para curvas PR pode expor diferenças de performance entre algoritmos não aparentes em curvas ROC.

Todavia alterar unicamente a métrica de performance, pode não ser suficiente para diminuir o efeito provocado, nos algoritmos, pelo desequilíbrio entre as classes. Técnicas complementares passam por metodologias ao nível dos dados, balanceando as distribuições ou removendo observações difíceis; ao nível dos algoritmos, aliviando o enviesamento em favor da classe maioritária; metodologias híbridas, que combinam as vantagens das duas anteriores (Krawczyk, 2016).

Kotsiantis, Kanellopoulos e Pintelas (2006) referem que metodologias, não aleatórias, ao nível dos dados, podem-se traduzir em melhores resultados, que metodologias ao nível dos algoritmos, porque fornecem nova informação ou eliminam a informação redundante que é apresentada ao algoritmo. Por sua vez, Weiss, McCarthy e Zabar (2007), referem que para conjuntos de dados inferiores a dez mil observações, metodologias ao nível dos dados realizando sobre amostragem, apresentam resultados superiores a metodologias híbridas**.**

Metodologias ao nível dos dados são realizadas através da realização de sob ou sobre amostragem, para reduzir o desequilíbrio entre as classes. Dentro desta família de metodologias as mais simples são a realização de sob ou sobre amostragem aleatória, que podem, respetivamente, descartar dados úteis ou aumentar a hipótese de o modelo realizar sobre ajustamento aos dados (López et al., 2013).

[3] Do inglês *Receiver operating characteristic curve (ROC)*

Assim, devido ao supracitado, a utilização de metodologias ao nível dos dados mais sofisticadas, não aleatórias, devem ser consideradas. A *Synthetic Minority Oversampling Technique* (SMOTE) é uma das abordagens mais reconhecidas para abordar o problema (López et al., 2013), realizando sobre amostragem, ao interpolar observações da classe minoritária que se encontram adjacentes, de forma a criar novas observações artificiais. Contudo como o SMOTE não considera a vizinhança quando gera novas observações, pode aumentar a sobreposição entre as classes (Wang & Japkowicz, 2004), degradando a capacidade do algoritmo em aprender independentemente do tamanho dos dados de treino (López et al., 2013). Prati, Batista e Monard (2004) demonstram mesmo que o grau de desequilíbrio entre classes não é o principal responsável por afetar a performance, mas sim o grau de sobreposição. Teoricamente qualquer problema "linearmente separável" pode ser resolvido independentemente da distribuição das classes (López et al., 2013).

Então outros algoritmos adaptativos, mais robustos, como o *Border-Line-SVM SMOTE* (Nguyen, Cooper, & Kamei, 2011) e o *Adaptative Synthetic Sampling* (ADASYN) (He, Bai, Garcia, & Li, 2008) foram propostos. O algoritmo ADASYN gera mais observações sintéticas para exemplos da classe minoritária que são mais difíceis de aprender, de forma adaptativa de acordo com a distribuição das classes, ao contrário do SMOTE que gera o mesmo número de observações sintéticas para cada amostra da classe. Por sua vez, o primeiro, foca-se na sobre amostragem das observações da classe minoritária que estão próximas da fronteira de decisão, partindo do princípio que estas serão mais importantes para a aprendizagem dos algoritmos.

Jo e Japkowicz (2004) e Kotsiantis et al. (2006) referem que, por vezes, independentemente do desequilíbrio entre as classes, os algoritmos de aprendizagem automática conseguem aprender os dados. Assim o desequilíbrio entre classes pode não ser o único problema a afetar a performance, pelo que resolver o problema do desequilíbrio isoladamente pode não aumentar a performance. Visa e Ralescu (2005) mencionam ainda que corrigir o desequilíbrio, numa proporção 50:50, alem de não garantir a melhoria da performance do algoritmo pode não ser mesmo a melhor proporção para treinar o modelo.

Além do já disposto, López et al. (2013) identificam e descrevem um conjunto de desafios, relevantes, e de elevada importância, que aparecem frequentemente associados a dados desequilibrados. Pequenos disjuntos[4] ocorrem quando a classe minoritária se encontra distribuída em vários pequenos aglomerados[5], sendo difícil identificar estas observações como relevantes ou ruído. Realizar *Boosting* dos algoritmos, ou usar métricas apropriadas como a precisão e a sensibilidade são técnicas para lidar com este problema. Alternativa passa pelo uso do método CBO (Jo & Japkowicz, 2004), que consiste em agrupar os dados de treino de cada classe separadamente, fazendo uso do algoritmo *K-Means*, realizando depois sobre amostragem aleatória em cada aglomerado. O ruído nos dados também pode prejudicar a aprendizagem da classe minoritária, que devido ao reduzido número de observações desta, pode influenciar a aprendizagem do conceito subjacente, pelo que uma abordagem de sobre amostragem que faça uso de um mecanismo de limpeza pode ser vantajosa, o algoritmo SMOTE+ENN (SENN) apresenta bons resultados nestes casos (Batista, Prati, & Monard, 2004), ao remover observações de ambas as classes, após sobre amostragem, caso estas sejam mal classificadas pelos seus vizinhos. *Dataset Shift* é definido como o caso onde os dados de treino e teste seguem diferentes distribuições, violando as assunções dos

[4] Do inglês *Small Disjuncts*
[5] Do inglês *Clusters*

algoritmos de classificação tradicionais. É um problema que pode afetar todo o tipo de problemas de classificação. Contudo, particularmente relevante, em dados desequilibrados, devido ao reduzido número de exemplos da classe minoritária. Ocorre geralmente devido a enviesamentos durante a amostragem na criação dos conjuntos de treino de teste. Em séries temporais dá-se o nome de deriva de conceito a problema similar, ou seja, quando os padrões do passado seguem diferentes distribuições dos padrões atuais.

Lidar, simultaneamente, com deriva de conceito, associado ao desequilíbrio dos dados assume assim importância. Ditzler e Polikar (2012) apresentam duas soluções que derivam do algoritmo Learn^{++}.NSE. A primeira solução, o Learn^{++}.CDS limita-se a incorporar o algoritmo SMOTE no processo, realizando sobre amostragem aos subconjuntos apresentados para treino dos diversos modelos. Por sua vez o algoritmo Learn^{++}.NIE cria subconjuntos de algoritmos através de *bagging* equilibrando ambas as classes. Os autores referem que ambos apresentam resultados, geralmente, superiores ao Learn^{++}.NSE, com vantagem para o Learn^{++}.NIE, no entanto este último recupera lentamente de mudanças abruptas no conceito, relativamente ao algoritmo tradicional Learn^{++}.NSE. Caso estas mudanças abruptas de conceito se verifiquem e sejam recorrentes nas séries financeiras, este algoritmo poderá apresentar reduzida performance.

2.3. SELEÇÃO DE VARIÁVEIS DE INDEPENDENTES

Além do anteriormente referido neste trabalho, na secção 2.1 e 2.2, a velocidade de classificação de um algoritmo é afetada pela quantidade de variáveis utilizadas. Ademais, devido ao valor de cada variável para o modelo não ser conhecido *a priori* é geralmente necessário começar com um número elevado de variáveis e proceder à sua redução gradual (Shivaji et al., 2013).

Assim a redução ou seleção das variáveis é considerada em termos da relevância e redundância para a variável dependente. Mais especificamente categorizadas em muito relevantes; pouco relevantes, mas não redundantes; irrelevantes e redundantes (Jovic, Brkic, & Bogunovic, 2015). Pelo que, dado um número finito de observações contendo variáveis redundantes, um algoritmo tradicional irá inicialmente diminuir o erro reportado para depois aumentar o mesmo, com o aumento continuado das variáveis consideradas para treino (De Silva & Leong, 2015).

Guyon e Elisseeff (2003) mostram que variáveis independentes irrelevantes, quando usadas em conjunto, com outras variáveis irrelevantes ou relevantes, podem-se revelar úteis. Estes autores mostram ainda que o uso de duas variáveis altamente, mas não perfeitamente, correlacionadas pode revelar-se melhor que o uso de unicamente uma das duas, porque correlação não é sinonimo de ausência de complementaridade entre ambas.

Essencialmente, existem três métodos para realizar a seleção de variáveis: *Wrappers*, Filtros e métodos Embutidos (Guyon & Elisseeff, 2003; Jovic et al., 2015).

Os w*rappers* utilizam um determinado modelo de aprendizagem automática para ordenar a importância das variáveis de acordo com o seu poder preditivo para o modelo. A performance do modelo é geralmente verificada num subconjunto de validação ou através de validação cruzada, e segundo (Jovic et al., 2015) obtêm-se subconjuntos de dados com maior capacidade preditiva que os obtidos utilizando filtros. Este método é geralmente implementado através de estratégias gananciosas e exaustivas, como busca para a frente ou eliminação para trás. Na primeira estratégia,

de forma inversa à segunda, as variáveis vão sendo progressivamente integradas no modelo de acordo com a sua contribuição para o resultado até não existir vantagem de adicionar nova variável. Por sua vez, nos métodos embutidos, o processo de seleção é interno ao algoritmo e otimizado para o mesmo.

Os filtros, são teoricamente menos exigentes computacionalmente que os *wrappers* (Guyon & Elisseeff, 2003), e como funcionam independentemente de um algoritmo de aprendizagem automática, são capazes de gerar melhor generalização aquando da aplicação de modelos posteriores, de acordo com o referido por Sánchez-Maroño et al. (2007), contrastando com o supracitado. A simplicidade e velocidade dos filtros tornam-nos, também, populares nos meios académicos e empresariais (Goswami, 2014). Dos filtros mais simples de aplicar, talvez seja o uso de medidas de correlação, tais como a de Pearson, Spearman, Informação Mutua ou mais recentemente a Distancia as mais utilizadas (Székely, Rizzo, & Bakirov, 2007). Escolhendo depois k variáveis, ordenadas pela ordem de importância reportada pelo filtro.

Alem dos referidos anteriormente baseados em medidas de correlação, existem vários filtros para realizar a seleção de variáveis. A família de filtros Relief é tolerante ao ruído e capaz de lidar com a interação entre as diferentes variáveis (Kira & Rendell, 1992), alem disso podem ser aplicados em todas as situações e possuem reduzido enviesamento (Sánchez-Maroño et al., 2007), contudo não ajudam a remover variáveis redundantes (Vina & Archana, 2014). Por sua vez, o filtro FCBF pode identificar e remover variáveis irrelevantes e redundantes (Vina & Archana, 2014), mas falha em lidar com as interações entre variáveis (Sánchez-Maroño et al., 2007). O filtro FCBF (L. Yu & Liu, 2003), considera a interdependência entre variáveis com respeito às classes , começando por considerar todas as variáveis e eliminando-as progressivamente com base na sua correlação, apresentando resultados competitivos com o filtro ReliefF.

Por sua vez, Peng et al.(2005) propõem o filtro mRMR, que pode tanto fazer uso de medidas de informação mútua, correlação ou distancia/semelhança entre variáveis para as selecionar, com o intuito de penalizar variáveis pela sua redundância quando na presença de outras.

Para dados desequilibrados, Chen e Wasikowski (2008) propõem o filtro FAST, obtendo resultados superiores, quando o comparam ao filtro ReliefF e a medidas de correlação tradicionais. Principalmente quando o número de variáveis selecionadas é reduzido e as classes são consideravelmente desequilibradas. Este algoritmo testa várias fronteiras de decisão, obtendo assim a ordenação das variáveis, fazendo uso da curva ROC. Jamali, Bazmara e Jafari (2012) confirmam os resultados anteriores testando diferentes algoritmos de seleção de variáveis em diferentes conjuntos de dados desequilibrados. Concluindo que os melhores métodos são o FAST, a relação sinal-ruído (SNR) que maximiza a diferença de médias entre duas classes minimizando a sua variação interna, e o *Information Gain* (IG). Sendo que o FAST se comporta melhor quando o conjunto de dados é altamente desequilibrado (5:1) e o IG quando os dados são moderadamente desequilibrados (3:1).

3. METODOLOGIA

Neste estudo testou-se a capacidade de diferentes modelos e metodologias para prever as séries financeiras referidas no anexo A. Para responder às questões anteriormente levantadas, testaremos três abordagens diferentes. Na primeira, e devido à bibliografia que o considera, partimos do principio de que os dados são i.i.d., na segunda abordagem respeitaremos a temporalidade dos dados no treino e na terceira utilizamos o modelo Learn^{++}.NEI, apropriado para dados desequilibrados, na presença de deriva de conceito. Para possuirmos base de comparação entre as três, reservamos o ultimo ano dos dados como teste, para realizar previsões.

De seguida, na secção 3.1, explicamos o pré-processamento aplicado aos dados.

3.1. DADOS E VARIÁVEIS

Os dados utilizados neste estudo foram obtidos do sítio *finance.yahoo.com*. Os dados obtidos são diários e compostos pelos valores mais alto e mais baixo verificado no dia, assim como os valores de abertura, fecho e volume transacionado. Além destes, os dados contêm o valor de fecho ajustado historicamente para desdobramentos de ações e dividendos. Este último é utilizado para treinar os modelos e ajustar as restantes variáveis para posteriormente derivar variáveis para o treino dos modelos. A fórmula 3.1 é utilizada em todas as variáveis, não ajustadas, exceto o volume onde se utiliza a fórmula 3.2 para obter valores ajustados historicamente.

$$PreçoAjustado_i = \frac{P.\,Ajust.\,Fecho_i}{P.\,Fecho_i} \times Preço_i \tag{3.1}$$

$$Volume\ Ajust_i = \frac{Volume_i \times P.\,Fecho_i}{P.\,Ajust.\,Fecho_i} \tag{3.2}$$

Consideramos 18 ações dos mercados da Alemanha, EUA e Portugal. Cada série contém dados compreendidos entre 1 de Janeiro de 2002 e 30 de setembro de 2017 (Anexo A). Os dados diários foram agrupados em semanas, de forma a reduzir o ruído. Algumas séries apresentam historicamente semanas para as quais estão em falta valores, pelo que imputamos linearmente estes casos . O último ano (1 de setembro 2016 a 30 setembro de 2017) foi reservado para teste, e os restantes para treino.

Utilizando os dados históricos agregados, os indicadores técnicos presentes no anexo J foram calculados, totalizando um total de 103 variáveis disponíveis para apresentar aos modelos. Com base nas variações do valor de fecho ajustado, as direções das séries foram, respetivamente, codificadas em 0 e 1, no caso de se apresentarem abaixo ou acima do limiar de variação de 5%, na janela de investimento considerada. Ao contrario da restante literatura que tenta prever movimentos que considera um limiar de 0%. Neste estudo consideramos como horizonte de investimento 4 semanas, ou seja fazemos previsões 4 semanas à frente. Horizontes mais reduzidos criam problemas consideravelmente desequilibrados, para a quantidade de observações, pelo que não foram considerados. Ademais, numa tentativa de reduzir o risco associado, só são classificadas como movimento de subida as variações que forem quase monotonicamente crescentes ao longo da janela de investimento futura, permitindo descidas até 1% numa semana intermédia.

3.2. MODELOS BASE UTILIZADOS

Nesta secção abordaremos, de forma não exaustiva, os algoritmos utilizados neste estudo e quais os híper-parâmetros que otimizamos. Caso não seja referido em contrario, utilizam-se os parâmetros dos modelos predefinidos na livraria *python* scikit-learn (v. 0.19.1.) (Pedregosa et al., 2012).

Uma SVM é um modelo de aprendizagem automática supervisionado, usado em problemas de classificação e regressão. Num problema de classificação uma SVM encontra o hiperplano que melhor segrega entre as classes. Nos casos em que os dados não são separáveis linearmente, o algoritmo permite utilizar diferentes funções (ou *kernel*) que permitem esta separação, de acordo com os padrões nos dados. Neste trabalho testamos como híper-parâmetros do modelo quatro funções (*kernel*) diferentes, a Linear, a Polinomial, a Sigmoide e a RBF. Um parâmetro importante do algoritmo é o valor de C, para valores elevados deste o algoritmo tende a ajustar-se melhor às observações, mas geralmente à custa da generalização fora da amostra. Neste estudo otimizamos $C \in [0.5, 1000]$.

As ANN consideradas neste estudo, são *Multi-layer Perceptron* (MLP) e como tal alimentadas para a frente, o que significa que os nós da rede estão totalmente conectados aos nós na camada anterior mas não conectados a nós na própria camada. Pelo que só as saídas dos nós nas camadas estão conectados às entradas dos nós na camada seguinte. Neste trabalho testamos redes com até 3 camadas escondidas e até 50 nós por camada, nos nós escondidos testamos de entre três funções de ativação distintas, a linear retificada ReLU (3.3), a função tangente hiperbólica (3.4) e a função sigmoide (3.5).

$$\varphi(x) = max(0, x) \tag{3.3}$$

$$\theta(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.4}$$

$$\lambda(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$

A velocidade de aprendizagem é também um hiperparâmetro que otimizamos, porque esta pode depender da estrutura das camadas e de outros híper-parâmetros. Como não permitimos paragem prematura do treino do MLP, para possuirmos maior numero de dados para treino, definimos um numero de *epochs* máximo permitido, que também é otimizando para valores até 1500.

No algoritmo KNN o principio subjacente ao método é o de encontrar um numero pré-determinado de K observações semelhantes à observação para a qual se pretende obter a estimação, normalmente, e utilizada neste estudo, através da distancia euclidiana. Determinar o valor ótimo para K assume então destaque. Neste trabalho otimiza-se para valores de $K \in [2, 30]$.

A *Logistic Regression (LOG)* estima a probabilidade de determinada característica binária estar presente, dado um conjunto de variáveis explicativas, estimando os coeficientes β de uma regressão do tipo 3.6, escolhendo coeficientes que maximizem a probabilidade.

$$logit(p) = \beta_0 + \beta_1 X_1 + \ldots \beta_n X_n \quad (3.6)$$

Neste trabalho, nós utilizamos uma abordagem regularizada deste algoritmo, que tenta evitar sobre ajustamento. Esta técnica sacrifica o enviesamento em troca de uma redução da variância dos valores previstos, melhorando, teoricamente, a performance. Contudo para tal é necessário otimizar um parâmetro C, onde da mesma forma que nas SVM, valores mais pequenos se traduzem numa regularização mais forte. Neste estudo otimizamos $C \in [0.2, 500]$

As RF são modelos compostos por múltiplas árvores de decisão. Implementam uma técnica chamada de *Bagging* ou agregação *Bootstrap* (Breiman, 1996) para produzir uma amostra dos dados individual para cada árvore de decisão, selecionando também um conjunto semialeatório de variáveis, pela razão de que se determinada variável possuir elevada capacidade preditiva será selecionada para múltiplas árvores, fazendo com que as árvores sejam correlacionadas, diminuindo a sua utilidade. Para cada observação, cada árvore vota individualmente numa classe, e a floresta reporta como previsão a classe que possui a maioria dos votos. Neste trabalho otimizamos a profundidade máxima permitida a cada árvore (entre 3 e 150), numa tentativa de evitar sobre ajustamento aos dados e o numero de árvores na floresta (entre 3 e 250).

Ke et al. (2017) propõem o algoritmo LightGBM (GBM), que similarmente ao anterior, utiliza árvores de decisão, contudo ao invés serem treinadas paralelamente e independentemente, são treinadas em sequencia. Aprendendo cada interação do algoritmo com os erros da anterior, ou seja realizando *boosting.* Otimizamos três paramentos do modelo. O parâmetro relativo ao "numero de folhas" máximo por árvore (entre 5 e 115), que controla a complexidade do algoritmo, sendo que se pode obter melhor performance para valores elevados arriscando sob reajustamento aos dados; para lidar com o sobre ajustamento otimizamos o parâmetro associado ao tamanho (profundidade) das árvores (entre 3 e 150). Por fim otimizamos também a velocidade e aprendizagem (entre 0.01 e 0.9).

Por sua vez o algoritmo *Gaussian Naive Bayes* (NB) possui como particularidade a velocidade elevada de treino por não possuir híper-parâmetros para otimizar, dado que não existem coeficientes que necessitam de ser otimizados. Unicamente a probabilidade de cada classe necessita de ser calculada. As probabilidades das classes são calculadas pela frequência da classe sobre o total das observações. Para prever as classes com base em variáveis quantitativas, o algoritmo assume que estas estão normalmente distribuídas (distribuição gaussiana), segmentando a variável condicionalmente à classe e calculando a média e desvio padrão da variável agrupada em ambas as classes, obtendo posteriormente probabilidades através da função densidade.

3.3. ALGORITMO PARA SELEÇÃO DO MELHOR MODELO

Neste trabalho pretende-se obter, automaticamente e com reduzida intervenção humana, o melhor modelo que maximize a previsão do correto movimento do mercado. Então dependendo das séries temporais consideradas ou dos diferentes momentos no tempo, distintos modelos podem apresentar diferentes performances.

Para efetuar a seleção dos modelos, incluindo a seleção de variáveis independentes e a otimização de híper-parâmetros, as observações podem ser separadas em subconjuntos de treino e teste (Guyon & Elisseeff, 2003). A este método dá-se o nome de *holdout,* contudo este método é um

estimador pessimista do erro fora da amostra, porque durante a seleção dos híper-parâmetros existe o risco de sobre-ajustamento do modelo aos dados de treino. Para resolver este problema um terceiro conjunto de dados chamado de validação pode ser reservado, diminuindo, no entanto, o número de observações disponíveis para treino. Este uso ineficiente dos dados torna-o menos correto para uso em amostras pequenas (Bengio & Grandvalet, 2004).

Devido a este problema, alternativa é o uso de validação cruzada (VC), que consiste em dividir o conjunto total de dados não reservados para teste em K subconjuntos, de treino e validação, mutuamente exclusivos e de tamanho aproximado, obtidos geralmente de forma aleatória ou estratificada (Figura 3.1). De acordo com Hastie, Tibshirani e Friedman (2009), Tsamardinos, Rakhshani e Lagani (2015), este é talvez o método mais comum para estimar a performance de um modelo em amostras pequenas e médias.

Figura 3.1. **Validação Cruzada em K Subconjuntos**

Figura 3.2. **Validação Sequencial em K subconjuntos com séries temporais**

Segundo Arlot e Celisse (2010) o uso de VC, como descrito na figura 3.1, evita o sobre ajustamento aos dados, porque a amostra de treino é independente da amostra de validação, considerando que os dados são independentes e identicamente distribuídos (i.i.d.). Neste estudo implementamos o algoritmo DOB-SCV (Moreno-Torres, Saez, & Herrera, 2012), para realizar a partição dos dados, que é uma forma robusta de garantir uma distribuição heterogénea das classes nas diferentes amostras, preservando a semelhança das distribuições evitando assim a introdução de *dataset shift* artificialmente durante a criação dos subconjuntos de treino e validação. Canedo, Moronõ e Betanzos (2015) concluem que em média o algoritmo DOB-SVC apresenta melhores resultados que a utilização da validação cruzada regular em múltiplos conjuntos de dados testados, incluindo conjuntos de dados com classes desequilibradas.

Contudo em dados temporais, o uso de VC é ligeiramente diferente porque as observações podem não ser independentes, devido a serem auto correlacionadas (De Silva & Leong, 2015), pelo que a adaptação para dados temporais fazendo uso de uma janela rolante, descrita pela figura 3.2, é proposta por Bengio e Chapados (2003) como alternativa.

O Algoritmo 1 utilizado, para seleção dos melhores modelos e híper-parâmetros, realiza a partição dos dados na linha 1 utilizando o algoritmo DOB-SCV ou utilizando a validação sequencial através de uma janela rolante. Realiza depois validação do modelo no ciclo da linha 17, calculando a performance de ajustamento na linha 31, quando realiza a média de todos os K conjuntos de validação. Neste estudo definimos $K=10$. Guardando a melhor iteração do algoritmo de otimização para o modelo entre as linhas 33 e 35.

Algoritmo 1 – seleção_melhor_modelo_VC()

```
TS_T, TS_V ← Particiona(TS) #Particionar a TS em K partições de treino e validação
ListaModelos ← Lista(SVM, MLP, etc....)
Para Modelo em ListaModelos fazer:
    Modelo_Hiper ← Inicializar lista de híper-parâmetros a otimizar

ErroModelos ← Lista() # de tamanho igual a comprimento de ListaModelos
Para Modelo em ListaModelos fazer:
    ErroTibshirani ← Lista(+Inf.) # de comprimento K, inicializada com valores infinitos
    Melhor_ajustamento ← +Inf.
    MelhorHiper ← Nulo

    Para opt=1 até limite fazer: #Inicializar algoritmo de otimização:

        H ← Hiperparametros(opt) #chamada ao algoritmo de otimização
        K_ajust, EntradasModelo ← Lista() # de comprimento K

        Para i=1 até K fazer: # Validação

            DadosTreino ← Normaliza(TS_T[i], H[funcaoNormalizacao])
            DadosValidacao ← Aplica_Normalizacao(TS_V[i], H[funcaoNormalizacao])

            EntradasModelo(i) ← SelecionarVariaveis(DadosTreino, H[nv], H[ms], H[ct])

            Se H[CorrigirDesiquilibrio] = Verdade então:
                EntradasModelo(i) ← CorrigirDesiquilibrio(EntradasModelo(i), H[mdesequilibro])

            ModeloTreinado ← Modelo(EntradasModelo(i), H)

            K_ajust(i) ← 1- PR_AUC(ModeloTreinado(DadosValidacao))

        E ← (1/K) Σ_{i=1}^{K} K_ajust(i)

        Se E < Melhor_ajustamento então:
            Melhor_ajustamento ← E
            MelhorHiper ← (H, EntradasModelo, K_ajust)

        Para i=1 até K fazer:
            Se ErroTibshirani(i) > K_ajust(i):
                ErroTibshirani(i) ← K_ajust(i)

    enviesamento ← (1/K) Σ_{i=1}^{K} ( ErroTibshirani[i] − MelhorHiper[K_ajust](i) )
    ErroModelos(Modelo) ← Melhor_ajustamento + enviesamento

Retorna MIN(ErroModelos)
```

Contudo a utilização de VC para efetuar o teste de múltiplos híper-parâmetros, reportando a performance dos melhores, traduz-se num resultado otimista da performance, que aumenta com a quantidade de modelos testados e com a redução do tamanho da amostra (Tsamardinos et al.,

2015). Para corrigir este problema e estimar o enviesamento da estimação do erro do modelo, obtido por VC, implementamos o método TT (Tibshirani & Tibshirani, 2009), que não adiciona complexidade ao processo, sendo portando computacionalmente eficiente quando comparado à alternativa proposta na literatura de utilizar *Nested cross-validation*[6]. Esta ultima técnica adiciona um ciclo externo ao ciclo de VC aumentando o numero de iterações de K para $K \times (K-1)$. Tsamardinos et al. (2015) comparam estas duas metodologias e concluem que ambas são equivalentes em termos dos resultados obtidos.

A correção do erro dos modelos pelo método TT é realizado, no final do processo de otimização, nas linhas 41 e 42. Utilizando o erro de ajustamento mínimo, dos vários híper-parâmetros testados no modelo, em cada partição, guardado durante as várias iterações do algoritmo de otimização, nas linhas 37 a 39.

Na linha 14, executamos o processo de otimização, para o qual utilizamos o algoritmo *Tree-structured Parzen Estimators* (Bergstra, Yamins, & Cox, 2013), implementado na livraria *python HyperOpt.* Neste estudo, limitado a 150 iterações, o algoritmo decide quais os híper-parâmetros a utilizar na iteração subsequente, de acordo com a performance das anteriores, começando contudo com um número predefinido de 20 inicializações aleatórias e possuindo a capacidade teórica de escapar a um mínimo local. Outros algoritmos como busca aleatória ou busca em grelha podem ser utilizados sem necessidade de alteração do procedimento utilizado.

Entre as linhas 19 a 22 realiza-se a normalização e seleção das variáveis, dentro do ciclo de VC. De acordo com Hastie et al. (2009) esta é uma condição necessária, para evitar que os modelos possuam a vantagem de obterem informação presente na amostra de validação através da seleção prévia de variáveis utilizando todos os dados disponíveis, fazendo com que a aplicação destes deixe de ser independente da amostra de validação. Exceção para os casos em que os algoritmos de seleção de variáveis sejam não supervisionados, ou por outras palavras, não possuam informação da variável dependente.

A linha 19 normaliza o conjunto de dados de treino utilizando diferentes funções, de acordo com o sugerido pelo algoritmo de otimização. Segundo Nayak, Misra e Behera (2014) diferentes funções de normalização podem afetar significativamente a performance dos algoritmos, sendo que nenhuma função, das testadas, se apresenta como superior. No corrente trabalho permitimos ao algoritmo de otimização escolher entre *z-score, minmax, madnorm, e thanhnorm* (Anexo I), descritas no estudo supracitado neste parágrafo. No entanto, devemos referir, que para certos algoritmos algumas das funções de normalização não são ótimas. Considere-se o caso de um MLP, cujas funções de ativação são tangentes hiperbólicas, neste caso o algoritmo é subótimo se os dados não estiverem normalizados no intervalo $[-1,1]$. Por uma questão de complexidade optamos por permitir ao algoritmo de otimização detetar estas ineficiências livremente, através do erro nas partições de validação, não limitando as opções de treino.

Na linha 22 do algoritmo 1 procede-se à seleção das variáveis que servirão para treinar o modelo, fazendo uso do algoritmo 2.

[6] Não foi possível identificar a fonte original para a metodologia. Tsamardinos et al. (2015) referem que as primeiras referencias, independentes, onde identificaram o seu uso, datam de 2005.

Nas linhas 24 e 25, corrige-se o desequilíbrio dos dados a apresentar ao modelo, permitindo ao algoritmo de otimização escolher se pretende ou não realizar a correção (linha 24). Novamente esta correção de desequilíbrio deve ocorrer dentro da VC, porque se corrigirmos o desequilíbrio dos dados fora da VC corremos o risco de validar o modelo em observações artificiais que contem informação derivada dos dados utilizados para treino. À semelhança da normalização, permitimos ao algoritmo de otimização selecionar de entre diferentes algoritmos para a realização de sobre amostragem, o ADASYN, SMOTE, SENN e *Border-Line-SVM* SMOTE, presentes na livraria *imbalanced-learn* (Lemaître, Nogueira, & Aridas, 2017). Neste trabalho, optamos por não otimizar os parâmetros dos algoritmos de sobre amostragem, devido ao aumento da complexidade computacional, assim obtemos proporções 1:1 nas amostras de treino através da aplicação dos algoritmos em cinco observações vizinhas.

Por fim, na linha 44, reajusta-se o melhor modelo com os respetivos híper-parâmetros no conjunto total de dados, seguindo o processo descrito entre as linhas 19 e 27, para realizar previsões para o futuro.

Algoritmo 2 – Entradas - DadosDeTreino, Num_Variaveis, FiltrosSelecao, corr_thresh

```
1:  Rank ← Matriz() #tamanho igual a ModelosSelecao X Numero de Variáveis nos Dados de Treino
2:
3:  i ← 0
4:  Para Filtro em FiltrosSelecao fazer:
5:      Rank[i] ← Filtro(DadosDeTreino) #guarda a classificação ordenada de todas as variáveis
6:      i ← i +1
7:
8:  Inter_rank ← Média(Rank) #Vetor de tamanho igual ao Numero de Variáveis nos Dados Treino
9:
10: i ← 1
11: Final_rank ← Lista()
12: Final_rank[0] ← Inter_rank[0] #adiciona 1ª variável na classificação média
13:
14: Para var em Inter_rank[1:] fazer:
15:     Adiciona ← Verdade
16:
17:     Para finalvar em Final_rank fazer:
18:         Se cov(var, finalvar)/σ_var σ_finalvar > corr_thresh:
19:             Adiciona ← Falso
20:
21:     Se Adiciona = Verdade:
22:         Final_rank[i] ← var
23:         i ← i +1
24:
25: Retorna Final_rank[0:Num_variaveis]
```

O algoritmo 2 permite combinar diferentes filtros utilizados para seleção de variáveis. Na linha 1 inicializa-se uma matriz para guardar o poder preditivo das diferentes variáveis (linhas 4 a 6) de acordo com os diferentes filtros considerados e previamente selecionados pelo processo de

otimização no algoritmo 1. O algoritmo de otimização pode escolher entre os filtros ReliefF, mRMR e FCBF, implementados na livraria *python* Scikit-Feature (Li et al., 2016) e os filtros SNR e FAST, assim como qualquer combinação entre eles.

Seguidamente na linha 8 efetua-se a média dos resultados obtidos em cada filtro, para se obter um vetor que ordena a capacidade preditiva das diferentes variáveis.

Por fim, nas linhas 14 a 23, elimina-se deste vetor, as variáveis com menor poder preditivo, correlacionadas linearmente com variáveis com maior poder preditivo, de acordo com determinado limite de correlação proveniente do processo de otimização do algoritmo 1 (linha 18). Retornando na linha 25 um vetor de variáveis ordenadas por poder preditivo, mas também limitadas em número pelo processo de otimização do algoritmo 1. Permitimos que o limite de correlação, aplicado na linha 18, varie entre 0.60 e 0.99, porque, como referido anteriormente, correlação não é sinonimo de ausência de complementaridade entre variáveis.

3.4. DERIVA DE CONCEITO (LEARN^{++}.NIE)

Aprender num ambiente não estacionário requer que o algoritmo aprenda conceitos que mudam com o tempo.

Neste trabalho implementámos o algoritmo Learn^{++}.NIE que é uma abordagem que aplica uma técnica que permite lidar com o desequilíbrio entre classes, sem produzir dados sintéticos, ou recolher mais dados da classe minoritária, recorrendo à técnica de *Bagging* ou agregação *Bootstrap*, que no caso deste algoritmo dá ênfase à seleção da classe minoritária equilibrando-a com a classe maioritária através uma seleção aleatória e com reposição. Produzindo desta forma conjuntos de vários modelos cuja agregação final é obtida por voto maioritário. A figura 3.3 demonstra a lógica subjacente ao algoritmo que considera N janelas de tamanho T .

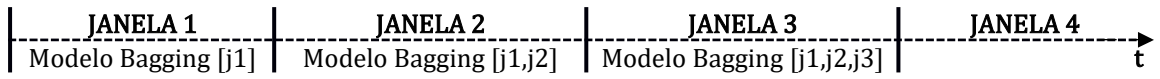


Figura 3.3. **Exemplificação da lógica subjacente ao algoritmo Learn++.NIE**

Na primeira Janela o algoritmo ajusta o Modelo Bagging [j1] através da combinação de R modelos agrupados por voto maioritário, neste trabalho $R=3$. Na Janela 2 o algoritmo ajusta tal como na janela 1 o modelo Bagging [j2], que posteriormente através de combinação ponderada com o modelo Bagging [j1] resulta no modelo Bagging [j1,j2]. Para tal o algoritmo Learn^{++}.NIE atribui pesos a cada um dos conjuntos de modelos Bagging [j1] e Bagging [j2], através do cálculo do erro de previsão de ambos na janela 2, e aplicando a fórmula 3.7 como medida de performance, resultando assim no modelo Bagging [j1,j2].

$$\varepsilon_k = \eta\left(1-\frac{tp}{tp+fn}\right)+(1-\eta)\left(1-\frac{tn}{tn+fp}\right) \tag{3.7}$$

Onde *tp* e *tn* correspondem ao numero de verdadeiros positivos e negativos, enquanto *fn* e *fp* ao numero de falsos negativos e positivos, respetivamente. A formulação matemática para

posterior cálculo dos pesos, de cada modelo pode ser consultada no trabalho original dos autores (Ditzler & Polikar, 2012).

Outras métricas sugeridas pelos autores em substituição da 3.7, mas desconsideradas neste estudo, são a *F-Score* e a *G-Mean.*

Assim o modelo Bagging [j1] fará previsões não enviesadas para a Janela 2, o Bagging [j1,j2] para a Janela 3 e o Bagging [j1,j2,j3] para a janela 4. Isto permite ao algoritmo recuperar ou descartar rapidamente padrões antigos assim como aprender novos para previsão futura, com base neste recalculo constante dos pesos e adição de novos modelos ao sistema.

Porém surgem duas questões, relativamente ao ajustamento do modelo; nomeadamente qual o número ótimo de janelas a considerar e qual o valor de η para maximizar a performance. Relativamente a η os autores recomendam o uso do valor 0.5, quando não existe conhecimento prévio do problema. Encontrar o número ótimo de janelas, por outro lado, depende dos potenciais padrões de deriva de conceito que existem nos dados, e como tal exige o teste de vários valores para N , dentro da amostra. Neste trabalho testamos valores $2 \leq N \leq 15$.

Além do referido, esta metodologia utiliza como algoritmo base determinado modelo. Neste trabalho limitamo-nos à utilização dos modelos descritos na secção 3.2. No entanto permanece a questão de quais os melhores híper-parâmetros para treinar o modelo e quais as melhores variáveis a selecionar. Desta forma ao invés de utilizarmos um qualquer algoritmo base, utilizamos a metodologia descrita na secção 3.3 para selecionar o melhor modelo para cada subagrupamento. Devido ao limitado número de dados contidos em cada janela que são ainda posteriormente divididos por diversos conjuntos de validação, particionamos os dados utilizando o algoritmo DOB-SCV, para maximizar o numero de dados disponíveis para treino e minimizarmos a hipótese de possuirmos unicamente dados de uma única classe para treino. No entanto estamos a violar a hipótese de os dados serem independentes, que ignoramos de forma informada, uma vez que as amostras *Bootstrap* tenderão a atenuar este problema. Esta abordagem para selecionar modelos, deverá melhorar a performance, uma vez que aumentará a diversificação nos subagrupamentos.

4. RESULTADOS E ANÁLISE

Nesta secção investigamos a capacidade de previsão das metodologias testadas. Para comparar a performance dos modelos de forma não enviesada é necessário testa-los em dados desconhecidos, ou seja, fora da amostra utilizada para treino dos modelos. Comparamos a performance dos modelos simultaneamente em todas as séries de modo a descobrir qual o modelo com melhor generaliza transversalmente a todas as séries testadas. Como anteriormente referido testamos um horizonte de previsão de 4 semanas. São consideradas cinco medidas para comparação da performance dos modelos, cujas formulas podem ser consultadas no Anexo G.

De entre todas as abordagens, a que ignora a hipótese de os dados possuírem uma componente temporal e observações dependentes, realizando validação cruzada e estratificando os dados através o método DOB-SCV, sobre ajusta aos dados de treino em grande parte das séries, apresentando como tal performance inferior nos dados de teste (Anexo D). Este comportamento

deve-se provavelmente à violação da hipótese do dados serem i.i.d., nos dados utilizados para treino e validação, tal como referido noutros estudos (Arlot & Celisse, 2010; Roberts et al., 2017). É um resultado que pode ocorrer ou ser agravado pelo uso do método de partição DOB-SCV, que particiona os dados com base na distancia euclidiana entre observações, dividindo observações potencialmente auto correlacionadas pelas diferentes partições de validação, fazendo com que estas não sejam independentes entre elas. Pelo que se revela menos útil ou mais arriscada para uso em investimento. Exceção feita quando a metodologia seleciona o modelo LOG como melhor modelo, que devido ao facto de ser regularizado, é mais resistente ao sobre ajustamento. Contudo, devido ao referido, esta metodologia será desconsiderada do resto da análise.

Entre as duas restantes estratégias, na amostra reservada para teste, não existem diferenças estatisticamente significativas, em nenhuma medida utilizada para comparar a sua performance. Nos dados de treino as estratégias não são comparáveis, dado que a Learn^{++}.NIE é incremental com o tempo, adicionando novos modelos conjuntamente. Os resultados dentro da amostra são assim, obtidos seguindo diferentes janelas nos dados à medida que novos modelos vão sendo adicionados e ponderados no algoritmo, e úteis unicamente para selecionar o numero ótimo de janelas a utilizar no algoritmo Learn^{++}.NIE. Desta forma o ultimo conjunto de modelos não pode ser ajustado a todos os dados históricos, ao contrario da metodologia que faz validação sequencial (doravante TS VC). Por outro lado comparar os algoritmos na amostra de treino não é garantia de performance futura do algoritmo.

Para a comparação de dois algoritmos diferentes, em múltiplos conjuntos de dados, (Demšar, 2006) recomenda a utilização do teste de Wilcoxon, em substituição do teste T, dado que o primeiro é não paramétrico, resistente a valores extremos e não assume a normalidade dos dados, sendo então estatisticamente mais seguro. A tabela 4.1 reporta os resultado da comparação entre ambos, na amostra de teste.

As métricas sensibilidade e *F-Score* são as que mais diferem entre ambos os modelos, no entanto não suficiente para rejeitarmos a hipótese nula do teste de Wilcoxon, considerando um nível de confiança de 95%, pelo que neste cenário consideramos que o algoritmo Learn^{++}.NIE é estatisticamente equivalente, ao TS VC. Na verdade, algoritmo Learn^{++}.NIE maximiza internamente uma ponderação entre a sensibilidade de ambas as classes maioritária e minoritária, na ponderação que dá aos agrupamentos de modelos para realizar previsões.

Tabela 4.1.

Valores p para as medidas de performance comparando as metodologias TS VC e Learn^{++}.NIE na amostra de teste

	Precisão	Sens.	Cohen Kappa	Auc-PR	F-*Score*
Wilcoxon	0.51	0.11	0.21	0.43	0.10
Teste T	0.62	0.15	0.34	0.76	0.10

Nota: Considera-se o valor *p* significante a um nível $p < 0.05$. Sens. refere-se à métrica Sensibilidade. Auc-PR refere-se à área debaixo da curva Precisão-Revocação.

De acordo com os testes anteriores, ambos os algoritmos podem ser utilizados igualmente para negociar no mercado, dado que a sua performance em prever movimentos maiores que 5% é estatisticamente a mesma, contudo devemos também considerar a diferença no custo

computacional entre ambos. O algoritmo Learn^{++}.NIE, utilizando a metodologia descrita na secção 3.3 para escolha do modelo base e a escolha do melhor número de janelas a considerar, é consideravelmente mais lento de treinar que a metodologia que faz unicamente uso da validação cruzada para séries temporais, TS VC. A titulo de exemplo para processar a metodologia descrita em 3.3 fazendo uso dos modelos descritos em 3.2 demoramos em média 1.5 horas por série, utilizando uma versão paralelizada do algoritmo 1 numa máquina com 32 núcleos lógicos a 3.10GHz. O Algoritmo Learn^{++}.NIE, terá de processar 15 janelas, no pior cenário computacional permitido.

Por outro lado, ambos apresentam valores médios de precisão abaixo de 50% (Anexos C e E), ou seja, preveem, fora da amostra, em média, mais falsos positivos que positivos. Interessa então avaliar, também, o risco de investir com ambas as metodologias, ignorando o custo de oportunidade, ou seja, o custo de investir num crescimento futuro entre 0% e 5% e não num superior a 5%. A figura 4.1 compara ambas.

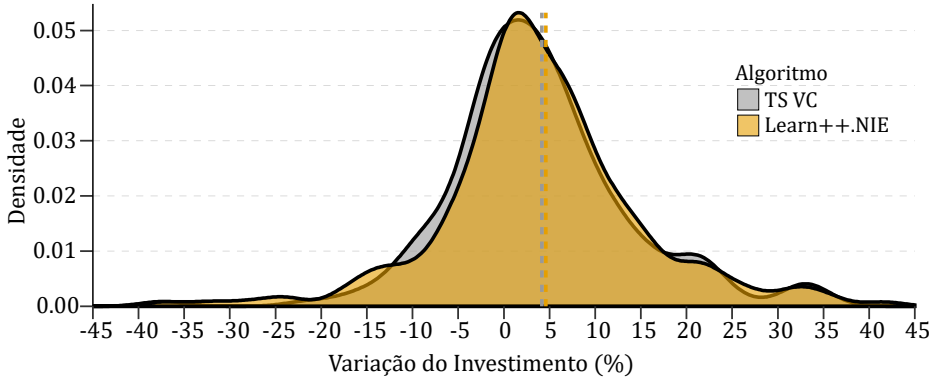


Figura 4.1. **Comparação fora da amostra para previsões na classe positiva.** O valor *p* para o teste T é de 0.7, não se rejeitando a Hipótese nula, considerando um nível de significância de 5%, de que não existe diferença entre as médias de ambas as distribuições. As linhas verticais tracejadas representam a média do algoritmo representado pela respetiva cor.

Da analise à figura 4.1 verificamos, que fora da amostra a distribuição de ambos os algoritmos se comporta similarmente, no que diz respeito à variação real do investimento. Ao calcularmos a integral da densidade para valores inferiores a zero, obtemos as probabilidades de ambas as estratégias resultarem em investimentos negativos. Ambas as abordagens apresentando probabilidades similares. Assim é possível inferir que apesar dos modelos errarem, em média, mais de 50% das vezes o risco associado a um investimento negativo reduz para 36% no caso do pior modelo, ainda que a probabilidade de errar seja de 59% (Tabela 4.2).

Tabela 4.2.

Probabilidade de variação do investimento fora da amostra, para previsões na classe positiva

Modelo	<0%	<5%	>=5%
TS VC	0.36	0.59	0.41
Learn++.NIE	0.34	0.56	0.44
TS VC Calibrado	0.20	0.35	0.65

Nota. A probabilidade é calculada através da integral da função densidade da variação real do investimento para todas as séries consideradas neste estudo, quando o modelo prevê uma subida superior a 5%, até ao limite definido na tabela (0% e 5%).

No entanto, para ambos os algoritmos, os resultados anteriores assumem que o melhor limiar probabilístico entre as classes é de 0.5. Contudo os modelos podem ser sob ou sobre confiantes relativamente às probabilidades que reportam. Ou seja, quando um modelo, possui as probabilidades bem calibradas, e reporta um valor de 0.85 para determinadas observações, então deveríamos estar confiantes de que 85% dessas observações pertencem à classe positiva (Caruana & Niculescu-Mizil, 2005). Nestes casos, a distribuição dos erros pode esconder incoerências nas previsões, que também não são facilmente detetados com medidas de performance tradicionais. A figura 4.2 demonstra as previsões realizadas pelo algoritmo Learn^{++}.NIE, evidenciando que as mesmas são sobre confiantes, principalmente para valores reportados com $p=1$, quando o algoritmo deveria estar 100% certo de uma subida superior a 5%, caso reportasse valores perfeitamente calibrados. Contudo estes valores reportados como probabilidades são simplesmente a classe prevista pela concordância dos diversos modelos base constituintes dos modelos de *bagging* treinados nas janelas, ponderados pelo peso associado à ultima janela, com valores normalizados entre 0 e 1. Ou seja, quando $p=1$ ou $p=0$ todos os modelos de *bagging* treinados nas janelas anteriores estão em concordância, excetuando o caso onde determinado modelo não concorde, mas possua um peso associado de zero.

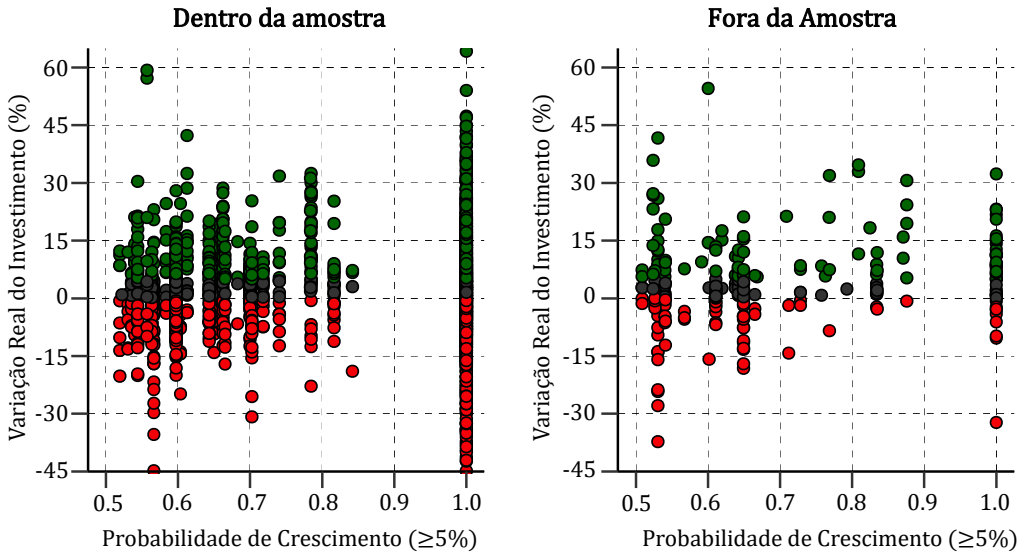


Figura 4.2. **Confiança das previsões dentro e fora da amostra para o algoritmo Learn++.NIE.** A verde estão representados os verdadeiros positivos. Os restantes pontos representam falsos positivos, a preto valores cujo retorno foi entre 0% e 5% e vermelho valores inferiores a 0%.

A figura 4.3 mostra o algoritmo TS VC, estatisticamente equivalente ao anterior, na variação do investimento e nas métricas de performance das diferentes séries, contudo denota-se que as probabilidades reportadas, embora não perfeitas, apresentam melhor calibração que as do anterior.

Desta forma melhorar a calibração das probabilidades dos modelos pode resultar num aumento da performance de uma estratégia de entrada no mercado apoiada em probabilidades, ao permitir o seu uso para negociar com maior confiança e melhor gestão do risco, devido a obter-se uma probabilidade mais próxima da real. Caruana e Niculescu-Mizil (2005) referem mesmo que alcançar modelos que reportem as verdadeiras probabilidades subjacentes, sem calibração, é difícil.

Neste estudo utilizamos a metodologia proposta por Kull et al. (2017), para calibrar os modelos que são possíveis de tal. Estes autores demonstram utilizando 41 conjuntos de dados que a abordagem proposta melhora geralmente os resultados de um modelo não calibrado, sendo mais eficaz que as técnicas tradicionais.

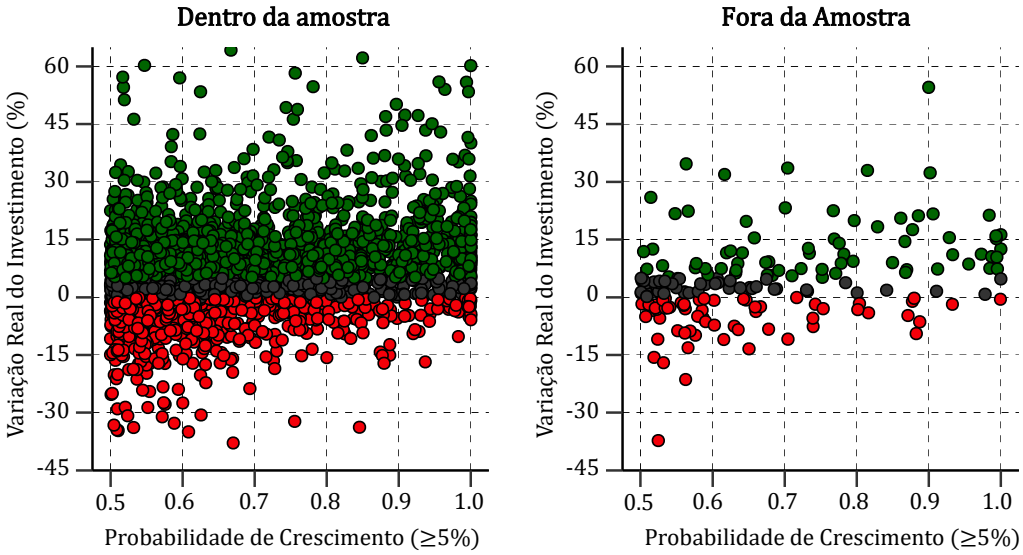


Figura 4.3. **Confiança das previsões dentro e fora da amostra para o algoritmo TS VC.** A verde estão representados os verdadeiros positivos. Os restantes pontos representam falsos positivos, a preto valores cujo retorno foi entre 0% e 5% e vermelho valores inferiores a 0%.

Para proceder à calibração das probabilidades dos modelos, geralmente guarda-se um conjunto de dados, independente da amostra de treino. Alternativa, que permite utilizar maior número de dados para treino, passa por fazer uso das previsões obtidas durante a validação cruzada/ sequencial nos conjuntos de validação nos dados de treino para calibrar o modelo. Contudo, esta ultima, torna ineficiente calibrar a metodologia DOB-SCV, por não restar informação para calibrar, devido ao perfeito ajustamento verificado nos dados de treino e validação. Por outro lado, a lógica subjacente ao algoritmo Learn^{++}.NIE, torna também difícil a aplicação desta calibração de probabilidades. Então calibramos, unicamente, o modelo TS VC para comparar aumentos de performance.

Para avaliarmos a qualidade das probabilidades antes e depois da calibração utilizamos a métrica de Brier, que avalia a qualidade das probabilidades reportadas pelo modelo, sendo que para um valor de zero as probabilidades são perfeitas. Na figura 4.4 avaliamos melhorias nesta métrica depois da calibração, subtraindo o valor da métrica ao seu valor no modelo não calibrado, permitindo assim avaliar a pertinência da calibração para a melhoria das probabilidades reportadas, concluindo que, geralmente, a calibração das probabilidades melhora ou não piora os valores reportados. Ao observarmos os valores da precisão é ainda possível observar um aumento desta métrica nas séries, ocorrendo devido a um compromisso entre a sensibilidade e a precisão, aumentando a ultima em detrimento da primeira (Anexos C e F). Contudo este resultado permite reduzir o risco associado a um erro de Tipo I.

Da figura 4.5 , na amostra de teste, verificamos que a calibração provoca uma redução do risco de investimento com variação negativa, ao observar-se um aumento da variação de investimento médio relativamente ao algoritmo não calibrado. A probabilidade de um investimento com variação negativa diminui para 20%, uma melhoria de 14p.p. face ao algoritmo Learn^{++}.NIE, por outro lado a probabilidade de se entrar no mercado para obter um retorno superior a 5% aumenta para 65% (Tabela 4.2).

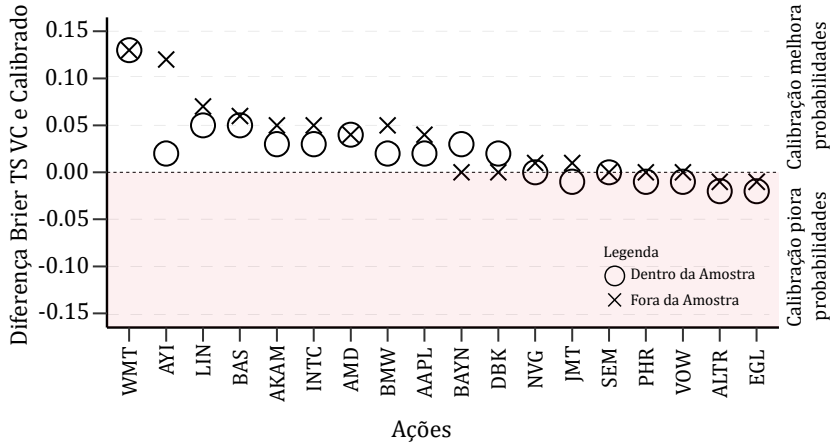


Figura 4.4. **Comparação da melhoria das probabilidades reportadas entre o modelo TS VC e o mesmo modelo calibrado.** Valores inferiores a zero significam que o modelo não calibrado apresenta melhores probabilidades que o modelo calibrado, na magnitude reportada no eixo vertical, de acordo com a métrica de Brier.

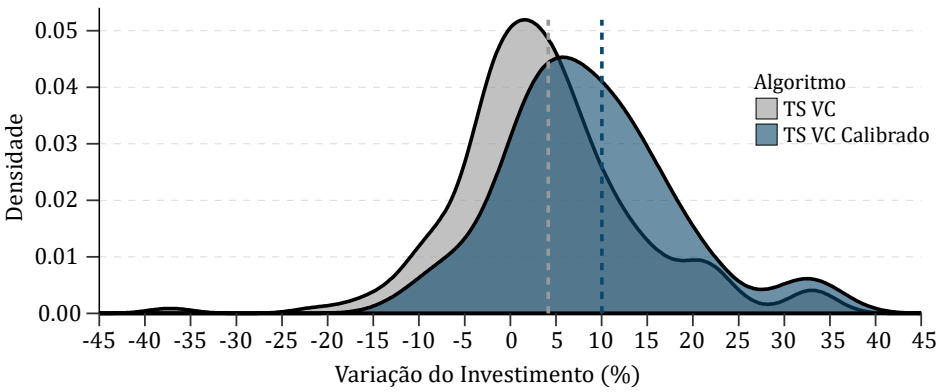


Figura 4.5. **Comparação das previsões na classe positiva na amostra de teste.** O valor *p* para o teste T é de 0.003, rejeitando-se a Hipótese nula, a um nível de significância de 5%, de que não existe diferença entre as médias de ambas as distribuições. As linhas verticais tracejadas representam a média do algoritmo representado pela respetiva cor. O Anexo H mostra a comparação dentro da amostra, com comportamento similar.

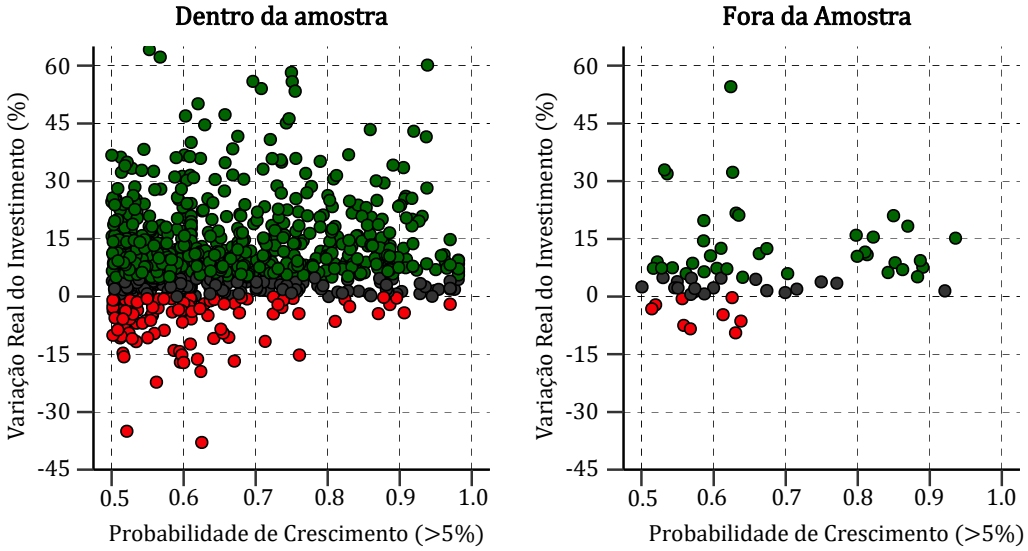


Figura 4.6. **Confiança das previsões dentro e fora da amostra para o algoritmo TS VC Calibrado.** A verde estão representados os verdadeiros positivos. Os restantes pontos representam falsos positivos, a preto valores cujo retorno foi entre 0% e 5% e vermelho valores inferiores a 0%.

Ora mas esta redução da sensibilidade pode levar a que o custo de oportunidade de não entrar no mercado, devido à redução de sinais, seja prejudicial. No entanto a redução do risco é clara e com a introdução de maior numero de séries no sistema espera-se que o numero de sinais aumente em conformidade.

5. CONCLUSÃO

Este estudo avaliou o uso de três abordagens distintas para a previsão de crescimentos superiores a 5% em 18 ações de três mercados diferentes. Calibrando posteriormente as probabilidades reportadas quando possível, reduzindo o risco da sua utilização, numa eventual estratégia de negociação

A abordagem que ignora a hipótese dos dados das séries serem auto correlacionados, considerando que os mesmos são i.i.d, não respeitando a temporalidade destes na amostra de treino sobre ajusta aos dados. Exceto quando os modelos utilizam regularização, apesar da técnica de validação cruzada. Estes resultados colocam em causa a generalização dos resultados obtidos nos estudos supracitados na secção 2.1, caso as amostras de treino e teste sejam obtidas de forma aleatória, sem respeito pela temporalidade dos dados.

Fazendo uso do algoritmo Learn++.NIE, o teste à hipótese de que os padrões entre as variáveis alteram com o tempo, inviabilizando ou dificultando a utilização de modelos tradicionais de aprendizagem automática, não se traduziu em resultados que sustentam a hipótese, quando comparado às abordagens alternativas utilizadas. No entanto não nos é possível rebater com confiança o referido por Harries e Horn (1995), Mitall e Kashyap (2016) relativamente à existência de deriva de conceito nas séries temporais, porque ao agregarmos os dados diários em semanais de forma a reduzir o ruído nos mesmos, podemos ter eliminado estas alterações de padrões com a

dimensão tempo, que podem ocorrer unicamente a um nível mais granular nos dados. Assim é pertinente no futuro testar também uma abordagem mais granular aos dados, devido à lacuna bibliográfica existente relativamente à utilização de abordagem preparadas para lidar com deriva de conceito em séries financeiras.

Por outro lado, nos dados e transformações consideradas, a complexidade computacional associada ao algoritmo Learn++.NIE, é desprovida de vantagem comparativamente aos algoritmos tradicionais. No caso deste trabalho não existe mesmo diferença devido à falta de evidencia estatística exigindo um nível de confiança de 95% quer para o teste de Wilcoxon, como para o teste T.

Considerar dados desequilibrados através da codificação de crescimentos acentuados para entrar no mercado revela-se também pertinente devido à redução de risco que se verifica. Ao provocar o deslocamento da média do erro de variação real do modelo de 0% para 5%, alguns dos erros associados ao modelo variarão entre 0% - 5%, traduzindo-se mesmo assim em investimentos com variação positiva, não obstante, mesmo um falso positivo com variação negativa apresenta valores mais próximos de zero devido à distribuição dos erros ser em torno de 5%. No entanto os resultados encontram-se alinhados com a performance reportadas na bibliografia discutida neste trabalho, para movimentos positivos ou negativos das séries, ou seja quando consideramos um limiar de variação de 0%. Apresentando a melhor metodologia obtida neste trabalho, a que respeita a temporalidade dos dados e possui probabilidades calibradas, uma probabilidade de prever corretamente movimentos superiores a 0% de 80%, fora da amostra. Apesar disto, para a metodologia proposta de seleção do melhor modelo, será pertinente de futuro testar diferentes variações e graus de desequilibro, de maneira a melhor entender o impacto deste tipo de codificação no resultados obtidos.

De referir ainda que no caso de ser importante minimizar o risco, ou seja reduzir a probabilidade de ocorrência de um erro Tipo I, calibrar os modelos traduz-se num aumento da sua precisão à custa de uma redução da sensibilidade. Esta estratégia para alguns modelos reduziu a sensibilidade do modelo a zero para determinadas séries fazendo com que estas não gerassem sinais, e que o numero de sinais das restantes fosse reduzido relativamente à abordagem não calibrada. No entanto, se ao invés de considerarmos 18 séries no sistema, considerarmos um numero consideravelmente maior de séries (assumindo generalização dos resultados), possuiremos um numero de sinais suficientes para entrar no mercado todas as semanas, apoiados num nível de risco inferior.

Contudo, importa também analisar a metodologia num ambiente de investimento neutro ou desfavorável. No período de teste o mercado financeiro apresentou uma tendência de subida (Anexo B). Ou seja, existe a hipótese, de os algoritmos mesmo cometendo um erro, a tendência do mercado ser coadjuvante na redução do erro reportado. Para testar este cenário desfavorável, uma pré-seleção equilibrada de séries com evoluções não correlacionadas unicamente negativas ou neutras, pode ser efetuada e testada.

Um outro caminho de análise que interessa investigar é a influencia da volatilidade nos algoritmos. Se a volatilidade for aleatória o algoritmo poderá ser prejudicado pela inclusão destas séries no sistema, por outro lado, caso tal não se verifique, uma maior volatilidade poderá ser vantajosa devido ao possível aparecimento de um maior numero de sinais.

Relativamente a melhorias no algoritmo e processo descrito para seleção dos melhores modelos num ambiente desequilibrado, diversos caminhos podem ser considerados, de forma a testar melhorias no mesmo. Ao invés de utilizarmos filtros podemos considerar métodos de decomposição lineares e não lineares para extrair novas variáveis para treino dos modelos, reduzindo a dimensionalidade, e substituindo desta forma o Algoritmo 2. Por outro lado ao invés de realizarmos sobre amostragem podemos realizar *bagging* dos modelos N vezes com reposição equilibrando as classes minoritária e maioritária, ou considerar diferentes proporções na realização da sobre amostragem.

Outra alternativa com potencial para melhorar as previsões finais, passa por realizar agrupamentos dos diversos modelos testados ou *staking,* utilizando um novo modelo de aprendizagem automática para aprender das previsões dos K conjuntos de validação dos diversos modelos testados, ignorando nestes caso o método TT. Esta abordagem adicionaria reduzida complexidade ao processo dado que os modelos base ao processo de *staking* ou agrupamento encontram-se todos treinados, devido a terem sido testados pelo método TT.

6. BIBLIOGRAFIA

Alavi, S. E., Sinaei, H., & Afsharirad, E. (2015). Predict the trend of stock prices using machine learning techniques. *International Academic Journal of Economics*, *2*(12), 1–11.

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2015). Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and its Applications*, *7*(3), 176–204.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*(0), 40–79. https://doi.org/10.1214/09-SS054

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations*, *6*(1), 20–29. https://doi.org/10.1145/1007730.1007735

Bengio, Y., & Chapados, N. (2003). Extensions to Metric-Based Model Selection. *Journal of Machine Learning Research*, *3*, 1209–1227. https://doi.org/10.1162/153244303322753634

Bengio, Y., & Grandvalet, Y. (2004). No Unbiased Estimator of the Variance of K-Fold Cross-Validation Yoshua. *Journal of Machine Learning Research*, *302*(4), 1089–1105.

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a Science of Model Search : Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Em *Presented at the 30th International Conference on Machine Learning (ICML 2013)* (pp. 115–123).

Bifet, A., Morales-bueno, R., Baena-Garcia, M., Campo-Avila, J. Del, Fidalgo, R., Bifet, A., … Morales-bueno, R. (2006). Early Drift Detection Method. Em *4th ECML PKDD International Workshop on Knowledge Discovery from Data Streams* (Vol. 6, pp. 77–86). https://doi.org/10.1.1.61.6101

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Caruana, R., & Niculescu-Mizil, A. (2005). Predicting good probabilities with supervised learning. *Proceedings of the 22nd international conference on Machine learning - ICML '05*, 625–632. https://doi.org/10.1145/1102351.1102430

Chen, X., & Wasikowski, M. (2008). FAST: a roc-based feature selection metric for small samples and imbalanced data classification problems. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, 124--132. https://doi.org/10.1145/1401890.1401910

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. Em *Proceedings of the 23rd international conference on Machine learning - ICML '06* (pp. 233–240). https://doi.org/10.1145/1143844.1143874

De Silva, A. M., & Leong, P. H. W. (2015). *Grammar-based feature generation for time-series prediction*. *SpringerBriefs in Applied Sciences and Technology*. https://doi.org/10.1007/978-981-287-411-5

Degutis, A., & Novickytė, L. (2015). The Efficient Market Hypothesis: A Critical Review of the Literature. *IUP Journal of Financial Risk Management*, *12*(4), 48–63.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, *7*, 1–30. https://doi.org/10.1016/j.jecp.2010.03.005

Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). Forecasting to Classification : Predicting the direction of stock market price using Xtreme Gradient Boosting, (October), 1–10. https://doi.org/10.13140/RG.2.2.15294.48968

Ditzler, G., & Polikar, R. (2012). Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *25*(10), 2283–2301. https://doi.org/10.1109/TKDE.2012.136

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78. https://doi.org/10.1145/2347736.2347755

Dumasia, N. A., & Shah, A. (2016). Review Paper on Concept Drift in Process Mining. *International Research Journal of Engineering and Technology (IRJET)*, *3*(4), 1675–1678.

Elkan, C. (2001). The foundations of cost-sensitive learning. Em *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 973–978). https://doi.org/doi=10.1.1.29.514

Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *22*(10), 1517–1531. https://doi.org/10.1109/TNN.2011.2160459

Fama, E. F. (1965). The Behavior of Stock-Market Prices. *Journal of Business*. https://doi.org/10.1017/CBO9781107415324.004

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with Drift Detection. Em *Intelligent Data Analysis* (Vol. 3171, pp. 526–535). https://doi.org/10.1007/978-3-540-28645-5_29

Goswami, S. (2014). Feature Selection : A Practitioner View. *Information Technology and Computer Science*, *11*(October), 66–77. https://doi.org/10.5815/ijitcs.2014.11.10

Grossman, S. (1975). On the Efficiency of Competitive Stock Markets Where Trades Have Diverse Information. *Journal of Finance*, *31*(2), 573–585.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027

Hanchuan Peng, Fuhui Long, C. D. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

Harries, M., & Horn, K. (1995). Detecting concept drift in financial time series prediction using symbolic machine learning. *Ai-Conference*, 91–98.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction.* (2.ª ed.). New York: Springer. https://doi.org/10.1198/jasa.2004.s339

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Em *Proceedings of the International Joint Conference on Neural Networks* (pp. 1322–1328). https://doi.org/10.1109/IJCNN.2008.4633969

HE, H., & Garcia, E. a. (2010). Learning from Imbalanced Data Sets. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263--1264. https://doi.org/10.1109/TKDE.2008.239

Hoens, T. R., Polikar, R., & Chawla, N. V. (2012). Learning from streaming data with concept drift and imbalance: an overview. *Progress in Artificial Intelligence*, *1*(1), 89–101. https://doi.org/10.1007/s13748-011-0008-0

Jamali, I., Bazmara, M., & Jafari, S. (2012). Feature Selection in Imbalance data sets. *International Journal of Computer Science (IJCSI))*, *9*(3), 42–45.

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, *6*(1), 40. https://doi.org/10.1145/1007730.1007737

Jovic, A., Brkic, K., & Bogunovic, N. (2015). A review of feature selection methods with applications. *Ieee*, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458

Kara, Y., Acar Boyacioglu, M., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*, *38*(5), 5311–5319. https://doi.org/10.1016/j.eswa.2010.10.027

Ke, G., Meng, Q., Wang, T., Chen, W., Ma, W., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems 30*, (Nips), 3148–3156. Obtido de http://papers.nips.cc/paper/6907-a-highly-efficient-gradient-boosting-decision-tree.pdf

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *Applied Mathematical Finance*. Obtido de http://arxiv.org/abs/1605.00003

Kira, K., & Rendell, L. (1992). The feature selection problem: Traditional methods and a new algorithm. Em *Aaai* (pp. 129–134). https://doi.org/10.1016/S0031-3203(01)00046-2

Kolter, J., & Maloof, M. (2007). Dynamic Weighted Majority : An Ensemble Method for Drifting Concepts. *Journal of Machine Learning Research*, *8*, 2755–2790. https://doi.org/10.1.1.140.2481

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets : A review. *GESTS International Transactions on Computer Science and Engineering*, *30*(1), 25–36. https://doi.org/10.1007/978-0-387-09823-4_45

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232. https://doi.org/10.1007/s13748-016-0094-0

Kull, M., De Menezes, T., Filho, S., Flach, P., Filho, T. S., & Flach, P. (2017). Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, *54*, 623–631. Obtido de

http://proceedings.mlr.press/v54/kull17a.html%5Cnhttp://proceedings.mlr.press/v54/kull17a/kull17a.pdf

Kumar, M., & Thenmozhi, M. (2005). Forecasting Stock Index Movement : a Comparision of Support Vector Machines and Random Forest. *Forest, Indian Institute of Capital Markets 9th Capital Markets Conference Paper.*, 1–16. https://doi.org/10.2139/ssrn.876544

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18*, 1–5. https://doi.org/http://www.jmlr.org/papers/volume18/16-365/16-365.pdf

Leung, M. T., Daouk, H., & Chen, A. S. (1999). Forecasting stock indices: a comparison of classification and level estimation models.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R., Tang, J., & Liu, H. (2016). Feature Selection: A Data Perspective. *arXiv preprint arXiv:1601.07996*. Obtido de http://featureselection.asu.edu/

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, *250*, 113–141. https://doi.org/10.1016/j.ins.2013.07.007

Madge, S. (2015). Predicting Stock Price Direction using Support Vector Machines.

Mittal, V., & Kashyap, I. (2016). Empirical Study of Impact of Various Concept Drifts in Data Stream Mining Methods. *International Journal of Intelligent Systems and Applications*, *8*(12), 65–72. https://doi.org/10.5815/ijisa.2016.12.08

Moreno-Torres, J. G., Saez, J. A., & Herrera, F. (2012). Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1304–1312. https://doi.org/10.1109/TNNLS.2012.2199516

Nayak, S. C., Misra, B. B., & Behera, H. S. (2014). Impact of Data Normalization on Stock Index Forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, *6*(2014), 257–269. https://doi.org/ISSN 2150-7988 Volume

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, *3*(1), 4. https://doi.org/10.1504/IJKESDP.2011.039875

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.1007/s13398-014-0173-7.2

Prati, R. C., Batista, G. E., & Monard, M. C. (2004). Class imbalances versus class overlapping: an analysis of a learning system behavior. Em *MICAI 2004: Advances in Artificial Intelligence* (pp. 312–321). https://doi.org/10.1007/978-3-540-24694-7_32

Provost, F. (2000). Machine learning from imbalanced data sets 101. Em *Proceedings of the AAAI'2000 Workshop on …* (p. 3). https://doi.org/10.1.1.33.507

Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE*, *11*(5), 1–11. https://doi.org/10.1371/journal.pone.0155133

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., … Dormann, C. F. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929. https://doi.org/10.1111/ecog.02881

Sánchez-Maroño, N., Alonso-Betanzos, A., & Tombilla-Snaromán, M. (2007). *Filter methods for feature selection–a comparative study*. *Intelligent Data Engineering and Automated Learning - IDEAL 2007* (Vol. 4881). Springer, Berlin, Heidelberg. https://doi.org/https://doi.org/10.1007/978-3-540-77226-2_19

Shivaji, S., James Whitehead, E., Akella, R., & Kim, S. (2013). Reducing features to improve code change-based bug prediction. *IEEE Transactions on Software Engineering*, *39*(4), 552–569. https://doi.org/10.1109/TSE.2012.43

Shostak, F. (1997). In Defense of Fundamental Analysis : A Critique of the Efficient Market Hypothesis. *The Review of Austrian Economics*, *2*(2), 27–45.

Street, W. N., & Kim, Y. (2001). A streaming ensemble algorithm (SEA) for large-scale classification. Em *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01* (Vol. 4, pp. 377–382). https://doi.org/10.1145/502512.502568

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *Annals of Statistics*, *35*(6), 2769–2794. https://doi.org/10.1214/009053607000000505

Tibshirani, R. J., & Tibshirani, R. (2009). A bias correction for the minimum error rate in cross-validation. *The Annals of Applied Statistics*, *3*(2), 822–829. https://doi.org/10.1214/08-AOAS224

Tsamardinos, I., Rakhshani, A., & Lagani, V. (2015). Performance-Estimation Properties of Cross-Validation-Based Protocols with Simultaneous. *Artificial Intelligence: Methods and Applications: 8th Hellenic Conference*, *24*(October). https://doi.org/10.1142/S0218213015400230

Vaiz, J. S., & Ramaswami, M. (2016). A Study on Technical Indicators in Stock Price Movement Prediction Using Decision Tree Algorithms. *American Journal of Engineering Research*, *5*(12), 207–212.

Verónica Bolón-Canedo, Noelia Sánchez-Maroño, A. A.-B. (2015). *Feature Selection for High-Dimensional Data* (1.ª ed.). Springer International Publishing. https://doi.org/10.1007/978-3-319-21858-8

Vina, A. K., & Archana, C. L. (2014). Feature Selection for High Dimensional and Imbalanced Data - A Comparative Study. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *3*(11), 3800–3804. Obtido de http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-11-3800-3804.pdf

Visa, S., & Ralescu, A. (2005). The Effect of Imbalanced Data Class Distribution on Fuzzy Classifiers - Experimental Study. Em *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.* (pp. 749–754). https://doi.org/10.1109/FUZZY.2005.1452488

Wang, B. X., & Japkowicz, N. (2004). Imbalanced Data Set Learning with Synthetic Samples. Em *Proceedings of IRIS Machine Learning Workshop*. Ottawa.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, *30*(4), 964–994. https://doi.org/10.1007/s10618-015-0448-4

Weiss, G., McCarthy, K., & Zabar, B. (2007). *Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? Dmin*. Obtido de http://storm.cis.fordham.edu/~gweiss/papers/dmin07-weiss.pdf

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, *23*(1), 69–101. https://doi.org/10.1007/BF00116900

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Em *International Conference on Machine Learning (ICML)* (pp. 1–8). https://doi.org/citeulike-article-id:3398512

Yu, X., Yu, M., Xu, L. X., Yang, J., & Xie, Z. Q. (2015). Training classifiers under covariate shift by constructing the maximum consistent distribution subset. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/302815

7. ANEXOS

Anexo A. Balanceamento das classes minoritária e maioritária nos dados

País	Ação	Amostra de Treino			Amostra de Teste		
		<5%	≥5%	Imb. rácio	<5%	≥5%	Imb. rácio
PORTUGAL	PHR	83%	17%	4.9	56%	44%	1.3
	ALTR	72%	28%	2.6	79%	21%	3.7
	SEM	79%	21%	3.7	71%	29%	2.5
	EGL	76%	24%	3.1	67%	33%	2.1
	JMT	76%	24%	3.2	83%	17%	4.8
	NVG	79%	21%	3.7	75%	25%	3
ALEMANHA	DBK	80%	20%	4	75%	25%	3
	BAYN	77%	23%	3.3	84%	15%	5.5
	BAS	75%	25%	3	77%	23%	3.3
	BMW	78%	22%	3.6	81%	19%	4.2
	VOW	72%	28%	2.5	79%	21%	3.7
	LIN	77%	23%	3.4	85%	15%	5.5
EUA	AMD	72%	28%	2.6	69%	31%	2.3
	AAPL	69%	31%	2.2	63%	37%	1.7
	AYI	70%	30%	2.3	83%	17%	4.8
	AKAM	74%	26%	2.8	83%	17%	4.8
	INTC	78%	22%	3.6	88%	12%	7.7
	WMT	88%	12%	7	87%	13%	6.4

Nota: Imb. rácio Representa o rácio de desequilibro entre ambas as classes, ou seja, o numero de observações da classe maioritária para uma da classe minoritária

Anexo B. Retorno das séries adquirindo a ação no inicio da amostra de teste e vendendo no final

País	Ação	Retorno do investimento
PORTUGAL	PHR	37.7%
	ALTR	17.82%
	SEM	32.91%
	EGL	40.98%
	JMT	22.38%
	NVG	30.41%
ALEMANHA	DBK	12,37%
	BAYN	19,67%
	BAS	17,52%
	BMW	6,84%
	VOW	2,79%
	LIN	10,11%
EUA	AMD	123,56%
	AAPL	61,88%
	AYI	-32,13%
	AKAM	-10,32%
	INTC	1,98%
	WMT	14,58%
	Média	**30,37%**

Anexo C. Medidas estatísticas para a previsão do modelo TS VC

			Resultados Dentro da Amostra							Resultados Fora da Amostra					
	série	Mod.	Auc PR	F1 Sco.	CK	Pr.	Se.	BS	TT	Auc PR	F1 Sco.	CK	Pr.	Se.	BS
EUA	AKAM	LOG	0.57	0.50	0.29	0.44	0.58	0.20	0.48	0.39	0.40	0.23	0.31	0.56	0.20
	AMD	LOG	0.54	0.53	0.33	0.48	0.59	0.21	0.53	0.64	0.45	0.13	0.38	0.56	0.23
	AYI	LOG	0.53	0.68	0.28	0.46	0.61	0.21	0.58	0.48	0.38	0.16	0.25	0.78	0.29
	AAPL	NB	0.52	0.44	0.25	0.55	0.36	0.22	0.59	0.73	0.33	0.21	0.80	0.21	0.25
	INTC	NB	0.42	0.43	0.23	0.37	0.51	0.20	0.63	0.24	0.18	0.09	0.20	0.17	0.16
	WMT	LOG	0.24	0.27	0.10	0.18	0.59	0.24	0.76	0.15	0.12	-0.12	0.07	0.29	0.25
ALEMANHA	BMW	LOG	0.57	0.58	0.45	0.51	0.68	0.16	0.64	0.50	0.38	0.22	0.36	0.40	0.18
	LIN	LOG	0.41	0.42	0.26	0.45	0.39	0.21	0.66	0.41	0.25	0.11	0.25	0.25	0.19
	VOW3	SVM	0.60	0.46	0.33	0.67	0.35	0.16	0.49	0.50	0.45	0.34	0.45	0.50	0.15
	DBK	LOG	0.48	0.52	0.38	0.44	0.64	0.17	0.69	0.65	0.15	0.02	0.48	0.77	0.29
	BAYN	MLP	0.55	0.53	0.36	0.48	0.59	0.18	0.57	0.69	0.57	0.51	0.67	0.50	0.09
	BAS	LOG	0.46	0.46	0.23	0.38	0.59	0.22	0.61	0.48	0.42	0.18	0.33	0.58	0.23
PORTUGAL	SEM	SVM	0.55	0.38	0.31	0.74	0.26	0.13	0.60	0.62	0.51	0.40	0.75	0.40	0.17
	EGL	LOG	0.55	0.38	0.29	0.73	0.25	0.15	0.61	0.53	0.11	0.08	1.00	0.06	0.21
	JMT	GBM	0.90	0.71	0.64	0.95	0.56	0.10	0.59	0.42	0.31	0.23	0.50	0.22	0.14
	PHR	MLP	0.53	0.24	0.20	0.80	0.14	0.11	0.75	0.46	0.15	0.02	0.50	0.09	0.29
	NVG	LOG	0.40	0.01	0.01	1.00	0.01	0.16	0.74	0.59	0.00	0.00	0.00	0.00	0.18
	ALTR	SVM	0.69	0.49	0.39	0.84	0.35	0.15	0.51	0.45	0.27	0.17	0.50	0.18	0.15
	Média		0.53	0.45	0.30	0.58	0.44	0.17	0.61	0.50	0.30	0.17	0.43	0.36	0.20

Nota: As métricas Precisão (Pr.) e Sensibilidade (Se.) correspondem à classe minoritária, enquanto a métrica F1 é uma média ponderada de ambas. Auc PR refere-se à área debaixo da curva Precisão-Revocação. CK é a métrica Cohen Kappa que é uma métrica apropriada para classes desequilibradas, comparando a exatidão do modelo contra a chance aleatória/exatidão expectável. Mod. Refere-se ao melhor modelo selecionado, dentro da amostra, de entre todos os testados para a série utilizando a metodologia descrita em 3.3. BS corresponde à métrica de Brier.

Anexo D. Medidas estatísticas para a previsão do modelo DOV-SCV

			Resultados Dentro da Amostra							Resultados Fora da Amostra					
	série	Mod.	Auc PR	F1 Sco.	CK	Pr.	Se.	BS	TT	Auc PR	F1 Sco.	CK	Pr.	Se.	BS
EUA	AKAM	LOG	0.66	0.52	0.40	0.71	0.41	0.14	0.43	0.42	0.43	0.35	0.60	0.33	0.13
	AMD	LOG	0.53	0.51	0.34	0.54	0.49	0.19	0.50	0.63	0.50	0.24	0.45	0.56	0.22
	AYI	GBM	0.99	0.96	0.94	0.95	0.96	0.05	0.44	0.25	0.34	0.10	0.22	0.78	0.34
	AAPL	RF	1.00	1.00	1.00	1.00	1.00	0.00	0.41	0.32	0.00	-0.10	0.00	0.00	0.30
	INTC	LOG	0.44	0.48	0.30	0.41	0.57	0.14	0.61	0.20	0.32	0.17	0.21	0.67	0.23
	WMT	MLP	1.00	0.94	0.93	0.90	0.98	0.01	0.71	0.09	0.00	-0.16	0.00	0.00	0.24
ALEMANHA	BMW	LOG	0.61	0.47	0.38	0.70	0.36	0.14	0.51	0.45	0.31	0.24	0.67	0.20	0.14
	LIN	MLP	1.00	0.98	0.97	0.99	0.96	0.02	0.54	0.29	0.45	0.32	0.36	0.62	0.18
	VOW3	SMV	1.00	1.00	1.00	1.00	1.00	0.04	0.41	0.17	0.10	-0.10	0.11	0.10	0.24
	DBK	GBM	1.00	1.00	1.00	1.00	1.00	0.01	0.57	0.52	0.40	0.27	0.57	0.31	0.16
	BAYN	GBM	1.00	1.00	1.00	1.00	1.00	0.00	0.45	0.58	0.40	0.36	1.00	0.25	0.10
	BAS	GBM	1.00	1.00	1.00	1.00	1.00	0.01	0.50	0.30	0.25	0.15	0.50	0.17	0.20
PORTUGAL	SEM	GBM	1.00	1.00	1.00	1.00	1.00	0.00	0.43	0.53	0.40	0.15	0.40	0.40	0.26
	EGL	RF	1.00	0.99	0.99	1.00	0.99	0.00	0.42	0.53	0.27	0.15	0.60	0.28	0.21
	JMT	LOG	0.51	0.50	0.30	0.42	0.62	0.20	0.56	0.41	0.38	0.26	0.43	0.33	0.17
	PHR	SVM	1.00	1.00	1.00	1.00	1.00	0.00	0.54	0.37	0.00	-0.11	0.00	0.00	0.47
	NVG	SVM	0.83	0.56	0.50	0.92	0.40	0.10	0.55	0.62	0.00	0.00	0.00	0.00	0.19
	ALTR	SVM	1.00	0.97	0.96	0.99	0.95	0.03	0.41	0.41	0.40	0.26	0.44	0.36	0.19
	Média		0.87	0.83	0.78	0.86	0.82	0.06	0.50	0.39	0.28	0.14	0.36	0.30	0.22

Nota: Ver nota do anexo C

Anexo E. Medidas estatísticas para a previsão do modelo Learn++.NIE (η = 0.5)

		Resultados Dentro da Amostra						Resultados Fora da Amostra					
	série	Auc PR	F1 Sco.	CK	Pr.	Se.	BS	Auc PR	F1 Sco.	CK	Pr.	Se.	BS
EUA	AKAM	0.56	0.35	0.25	0.67	0.23	0.21	0.58	0.46	0.40	0.75	0.33	0.14
	AMD	0.53	0.45	0.15	0.36	0.61	0.36	0.70	0.64	0.51	0.75	0.56	0.16
	AYI	0.51	0.34	0.19	0.53	0.25	0.23	0.68	0.42	0.21	0.28	0.89	0.24
	AAPL	0.57	0.50	0.16	0.40	0.64	0.38	0.47	0.29	0.07	0.44	0.21	0.27
	INTC	0.43	0.37	0.22	0.44	0.31	0.23	0.39	0.33	0.25	0.33	0.33	0.12
	WMT	0.29	0.25	0.14	0.24	0.27	0.15	0.27	0.18	0.09	0.25	0.14	0.13
ALEMANHA	BMW	0.52	0.48	0.31	0.44	0.53	0.28	0.48	0.40	0.26	0.40	0.40	0.15
	LIN	0.30	0.26	0.08	0.31	0.22	0.24	0.32	0.35	0.18	0.27	0.50	0.20
	VOW3	0.49	0.46	0.19	0.42	0.52	0.33	0.62	0.57	0.40	0.42	0.91	0.24
	DBK	0.45	0.40	0.23	0.36	0.44	0.25	0.53	0.56	0.42	0.58	0.54	0.21
	BAYN	0.41	0.35	0.11	0.31	0.41	0.27	0.39	0.38	0.26	0.38	0.38	0.14
	BAS	0.44	0.39	0.06	0.29	0.62	0.39	0.57	0.36	0.10	0.29	0.50	0.19
PORTUGAL	SEM	0.53	0.38	0.09	0.26	0.74	0.54	0.63	0.46	0.05	0.31	0.87	0.35
	EGL	0.41	0.37	0.04	0.26	0.66	0.49	0.58	0.60	0.36	0.52	0.71	0.20
	JMT	0.39	0.32	0.00	0.25	0.43	0.30	0.58	0.31	0.23	0.50	0.22	0.14
	PHR	0.35	0.29	0.12	0.24	0.36	0.25	0.59	0.56	-0.01	0.44	0.78	0.32
	NVG	0.32	0.23	0.07	0.32	0.17	0.27	0.58	0.16	0.27	0.57	0.31	0.27
	ALTR	0.52	0.38	0.23	0.56	0.29	0.25	0.35	0.17	0.14	1.00	0.09	0.20
	Média	0.44	0.37	0.15	0.37	0.43	0.30	0.51	0.40	0.23	0.47	0.48	0.20

Nota: Ver nota do anexo C. Não se reporta o melhor modelo porque vários diferentes podem ser selecionados para cada janela.

Anexo F. Medidas estatísticas para a previsão do modelo TS VC Calibrado

			Resultados Dentro da Amostra							Resultados Fora da Amostra					
	série	Mod.	Auc PR	F1 Sco.	CK	Pr.	Se.	BS	TT	Auc PR	F1 Sco.	CK	Pr.	Se.	BS
EUA	AKAM	LOG	0.57	0.44	0.33	0.68	0.33	0.17	0.48	0.39	0.44	0.33	0.44	0.44	0.15
	AMD	LOG	0.54	0.31	0.21	0.70	0.20	0.17	0.53	0.64	0.55	0.45	1.00	0.38	0.18
	AYI	LOG	0.58	0.22	0.15	0.81	0.13	0.19	0.58	0.48	0.33	0.19	0.33	0.33	0.16
	AAPL	NB	0.52	0.16	0.09	0.65	0.09	0.20	0.59	0.73	0.33	0.21	0.80	0.21	0.22
	INTC	NB	0.42	0.00	0.00	0.00	0.00	0.17	0.63	0.24	0.25	0.20	0.50	0.17	0.11
	WMT	LOG	0.24	0.00	0.00	0.00	0.00	0.11	0.76	0.15	0.17	0.06	0.20	0.14	0.14
ALEMANHA	BMW	LOG	0.57	0.24	0.19	0.81	0.14	0.14	0.64	0.50	0.33	0.29	1.00	0.20	0.13
	LIN	LOG	0.45	0.32	0.21	0.53	0.23	0.16	0.66	0.41	0.33	0.26	0.50	0.25	0.12
	VOW3	SVM	0.60	0.44	0.31	0.68	0.32	0.17	0.49	0.48	0.45	0.31	0.45	0.45	0.16
	DBK	LOG	0.48	0.00	0.00	0.00	0.00	0.15	0.69	0.67	0.00	0.00	0.00	0.00	0.16
	BAYN	MLP	0.55	0.44	0.34	0.66	0.33	0.15	0.59	0.71	0.50	0.44	0.75	0.38	0.09
	BAS	LOG	0.46	0.10	0.06	0.65	0.05	0.17	0.61	0.48	0.15	0.12	1.00	0.08	0.16
PORTUGAL	SEM	SVM	0.55	0.40	0.32	0.73	0.27	0.13	0.60	0.62	0.52	0.40	0.75	0.40	0.17
	EGL	LOG	0.55	0.02	0.02	1.00	0.01	0.17	0.61	0.53	0.00	0.00	0.00	0.00	0.22
	JMT	GBM	0.90	0.49	0.42	0.96	0.33	0.11	0.59	0.42	0.18	0.13	0.50	0.11	0.13
	PHR	MLP	0.53	0.13	0.11	0.83	0.07	0.12	0.75	0.47	0.00	0.00	0.00	0.00	0.29
	NVG	LOG	0.40	0.12	0.08	0.60	0.07	0.16	0.74	0.59	0.50	0.39	0.71	0.38	0.15
	ALTR	SVM	0.69	0.36	0.28	0.91	0.23	0.17	0.51	0.53	0.17	0.14	1.00	0.09	0.14
	Média		0.53	0.23	0.17	0.62	0.16	0.16	0.61	0.50	0.29	0.22	0.55	0.22	0.16

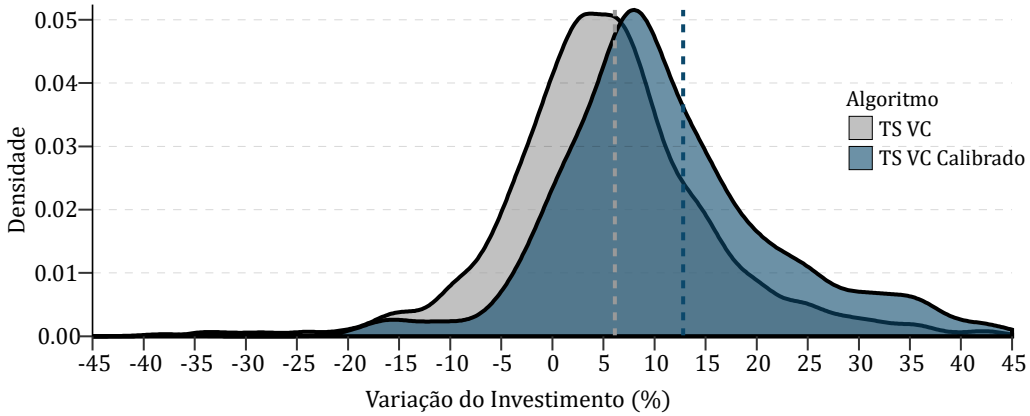
Nota: Ver nota do anexo C

Anexo G. Medidas de Performance

F1 Score	$2\times\left(\frac{tp}{tp+fp}\times\frac{tp}{tp+fn}\right)/\left(\frac{tp}{tp+fp}+\frac{tp}{tp+fn}\right)$
Cohen Kappa	$\frac{p_o-p_e}{1-p_e}$, com $p_e=\frac{1}{N^2}\sum_k n_{k1}n_{k2}$ e $p_o = exatidão$ Para k classes, N é o número de observações e n_{ki} o número de vezes que o algoritmo previu a classe ki
Precisão	$\frac{tp}{tp+fp}$
Sensibilidade ou Revocação	$\frac{tp}{tp+fn}$
Brier	$\frac{1}{N}\sum_{t=1}^{N}(f_t-o_t)^2$ Sendo f_t a probabilidade reportada pelo algoritmo e o_t a codificação da classe (0 ou 1)
Exatidão	$\frac{tp+tn}{tp+fn+fp+tn}$

Nota: *tp* corresponde ao numero de verdadeiros positivos, *fn* ao numero de falsos negativos, *tn* ao numero de verdadeiros negativos e *fp* ao numero de falsos positivos

Anexo H. Comparação das previsões na classe minoritária dentro da amostra



Nota: O valor *p* para o teste T é de 0.00, rejeitando-se a Hipótese nula, a um nível de significância de 5%, de que não existe diferença entre as médias de ambas as distribuições. As linhas verticais tracejadas representam a média do algoritmo representado pela respetiva cor

Anexo I. Formulas de normalização de dados

Z-score	$\frac{x_i - \mu(x)}{\sigma(x)}$ Onde $\mu(x)$ é o valor médio e $\sigma(x)$ é o desvio padrão
Min Max	$\frac{x_i - min(x)}{max(x) - min(x)}$
Mad Norm	$\frac{x_i - mediana(x)}{mediana(\lvert x_i - mediana(x) \rvert)}$
Tanh Norm	$\frac{1}{2} \times \left[\tanh \left[0.01 \frac{x_i - \mu(x)}{\sigma(x)} \right] + 1 \right]$

Anexo J. Indicadores técnicos utilizados como entradas nos modelos

RSI	$100-100/\left(1+\frac{\sum_{i=0}^{n-1} n^{o}\ Subidas_{t-i}}{n}/\frac{\sum_{i=0}^{n-1} n^{o}\ Descidas_{t-i}}{n}\right)$	N = 3, 5, 8, 10,12,14,16,18,20,25
CCI	$\frac{\left(\frac{H_t+L_t+C_t}{3}-\sum_{i=1}^{n}(H_{t-i+1}+L_{t-i+1}+C_{t-i+1})\right)}{\left(0.015\times\sum_{i=1}^{n}\lvert(\frac{H_{t-i+1}+L_{t-i+1}+C_{t-i+1}}{3})-\sum_{i=1}^{n}(H_{t-i+1}+L_{t-i+1}+C_{t-i+1})\right)}$	N = 3, 5, 8, 10, 12, 14, 16, 18, 20, 25
MA(n)	$\left(\sum_{i=1}^{n}\text{PreçoFecho}_{t-i+1}\right)\Big/n$	N = 3, 5, 8, 10,12, 14,16,18, 20, 25, 30, 35
MA_Cruzamentos	$MA(n)_t-MA(j)_t,\ j\in[5,10,15,20,25,30,35]\wedge n<j$	N = 3, 5, 8,10,12, 14,16,18, 20, 25, 30, 35
PVar(n)	$\left(\text{PreçoFecho}_{\text{t-n}}\text{-PreçoAbertura}_{\text{t-n}}\right)/\text{PreçoAbertura}_{\text{t-n}}$	N = 0,1,2
Mvar	$\sum_{n=0}^{2}\frac{\text{Pvar}(n)}{3}$	
PROC	$\frac{C_t-C_{t-n}}{C_{t-n}}$ C_t é o preço de fecho no momento atual	N = 3, 5, 8, 10, 12, 14, 16, 18, 20
%K	$100\times\frac{C_t-LO_n}{HO_n-LO_n}$ Sendo HO_n o valor mais alto nos últimos n dias e LO_n o valor mais baixo. C_t é o preço de fecho no momento atual	N = 3, 5, 8, 10, 12, 14, 16, 18, 20
%R	$-100\times\frac{HO_n-C_t}{HO_n-LO_n}$ Sendo HO_n o valor mais alto nos últimos n dias e LO_n o valor mais baixo. C_t é o valor de fecho no momento atual	N = 3, 5, 8, 10, 12, 14, 16, 18, 20
%D	$\sum_{i=0}^{n-1}\frac{\%K_{t-i}}{n}$	N = 3, 5, 8, 10, 12, 14, 16, 18, 20
A/O Oscilador	$\frac{HO_t-C_{t-1}}{HO_t-LO_t}$	
TR	$Tr_{t-1}-(Tr_{t-1}/n)+Tr_t$ Onde Tr é $max((H_t-L_t),(H_t-C_{t-1}),(C_{t-1}-L_t))$	N = 3, 5, 8, 10, 12, 14, 16, 18, 20
AverageDX	$\frac{ADX_{t-1}\times(n-1)+DX_t}{n}$	N = 3, 5, 8, 10, 12, 14, 16, 18, 20